CARL VON OSSIETZKY universität OLDENBURG

Department für Informatik
Abteilung Wirtschaftsinformatik

Projektabschlussbericht

2. Oktober 2010

Projektgruppe STORM

Inhaltsverzeichnis

I Einführung **9**

1 Einleitung **10**
1.1 Allgemeines zur Projektgruppe . . . 10
1.2 Motivation und Zielsetzung der Projektgruppe . . . 11
1.3 Lizenzbedinungen . . . 12
1.4 Aufbau des Projektabschlussberichts . . . 12

2 Projektorganisation **13**
2.1 Aufteilung in Entwicklergruppen . . . 13
2.2 Projektinterne Aufgabenbereiche . . . 14
2.2.1 Projektmanager . . . 14
2.2.2 Serveradministrator . . . 14
2.2.3 Testbeauftragter . . . 15
2.2.4 Homepage-Beauftragter . . . 15
2.2.5 Finanzbeauftragter . . . 15
2.2.6 Kommunikationsbeauftragter . . . 15
2.2.7 Layoutbeauftragter . . . 15
2.2.8 Dokumentationsbeauftragter . . . 16
2.3 Projekt Roadmap . . . 16
2.4 Vorgehensweise der Projektgruppe . . . 17
2.4.1 Methodische Vorgehensweise: Rapid Prototyping . . . 17
2.4.2 Technische Vorgehensweise: Continuous Integration . . . 19
2.5 Softwaretechnische Unterstützung im Projekt . . . 20
2.5.1 Serverseitige Software . . . 21
2.5.1.1 Organisations-Software . . . 21
2.5.1.2 Entwicklungssoftware i.w.S. . . . 22
2.5.2 Clientseitige Software . . . 24

3 Grundlagen zur Nachhaltigkeitsberichterstattung **26**
3.1 Was sind Nachhaltigkeitsberichte? . . . 26
3.2 Internetbasierte Nachhaltigkeitsberichterstattung . . . 26
3.3 Aufbau von Nachhaltigkeitsberichten . . . 27

II Anforderungsdefinition **28**

4 Vorbemerkungen zur Anforderungsdefinition **29**

5 Funktionale Anforderungen **30**
5.1 Systemadministration . . . 30
5.1.1 Rollenkonzept . . . 30

5.1.2 Beispiel-Konfiguration des Rollenkonzepts . . . 32
5.1.3 Anwendungsfälle im Bereich Benutzer- und Rechteverwaltung . . . 33
5.1.4 Registrierungsprozess . . . 33
5.1.5 Umgang mit Passwörtern . . . 33
5.1.6 Protokollierung . . . 33
5.1.7 An- und Abschaltbarkeit . . . 33
5.2 Schemaeditor . . . 36
5.2.1 Das Schema-Modell . . . 36
5.2.2 Schema verwalten . . . 36
5.2.3 Indikatoren verwalten . . . 37
5.2.4 Kategorien verwalten . . . 37
5.2.5 Optionale Anforderungen an den Schemaeditor . . . 37
5.3 Eingabeeditor . . . 38
5.3.1 Nachhaltigkeitsberichte verwalten . . . 38
5.3.2 Artikel verwalten . . . 39
5.4 Ausgabe von Nachhaltigkeitsberichten . . . 44
5.4.1 HTML . . . 44
5.4.2 PDF . . . 46
5.4.3 XML . . . 47
5.5 Usability-Funktionen . . . 47
5.5.1 Navigation . . . 47
5.5.2 Internationalisierung . . . 47
5.5.3 Validierung von Benutzereingaben . . . 47
5.6 Personalisierung . . . 49
5.6.1 Vorschlagswesen . . . 49
5.6.2 Fragen-System . . . 49
5.6.3 Template-Auswahl . . . 50
5.6.4 Auswertung von Benutzerprofilen . . . 50
5.7 Weitere Anforderungen . . . 51
5.7.1 Autovervollständigung von Suchbegriffen . . . 51
5.7.2 Bewertung von Inhalten . . . 51
5.7.3 Hilfefunktionen . . . 51
5.7.4 Kommentarfunktion . . . 51
5.7.5 RSS-Import . . . 51
5.7.6 Newsletter . . . 52
5.7.7 RSS-Feed . . . 52
5.7.8 TagCloud . . . 52
5.7.9 Trackback . . . 52
5.7.10 Startseite . . . 53

6 Nicht funktionale Anforderungen 54

III Entwurf 56

7 Vorbemerkungen zum Entwurf 57

8 Systemarchitektur 58
8.1 Server . . . 58
8.2 Datenbankmanagementsystem . . . 58

8.3 Grails . . . 59
8.4 Client . . . 60
8.5 Architektur-Überblick . . . 61
8.6 Komponenten . . . 62

9 Systemkomponenten 63
9.1 Systemadministration . . . 63
9.1.1 Modulverwaltung . . . 63
9.1.2 Rollenkonzept und Benutzerverwaltung . . . 65
9.1.3 Anwendungsfälle im Bereich Systemadministration . . . 69
9.2 Redaktionsbereich . . . 80
9.2.1 Schemaeditor . . . 80
9.2.2 Eingabeeditor für Berichte und Artikel . . . 83
9.3 Öffentlicher Bereich . . . 98
9.3.1 Darstellung der Berichte/Artikel . . . 98
9.3.2 Infokorb . . . 99
9.3.3 Kommentarfunktion . . . 101
9.3.4 Bewertungsfunktion . . . 102
9.3.5 Tags vergeben . . . 103
9.3.6 PDF-Generator . . . 103
9.3.7 XML . . . 104
9.3.8 RSS-Schnittstelle . . . 105
9.4 Recommender Engine . . . 107
9.4.1 Datenbasis . . . 107
9.4.2 Am häufigsten betrachtete Artikel . . . 107
9.4.3 Zuletzt betrachtete Artikel . . . 108
9.4.4 Ähnliche Artikel . . . 108
9.4.5 Artikel, die deine Follower betrachtet haben . . . 111
9.4.6 Artikel, die für deine Interessengruppe interessant sind . . . 111
9.5 Weitere Features . . . 112
9.5.1 TagCloud . . . 112
9.5.2 Autovervollständigung . . . 113
9.5.3 Sprachauswahl . . . 113
9.5.4 Template-Auswahl . . . 114
9.5.5 RSS-Import . . . 114
9.5.6 Newsletter . . . 114
9.5.7 Fragen-System . . . 115
9.5.8 Hilfe-System . . . 115
9.5.9 News . . . 115
9.5.10 Social Bookmarks/Networks . . . 115
9.5.11 Startseite . . . 116

IV Projektabschluss 117

10 Testbericht 118
10.1 Warum muss eine Webanwendung getestet werden? . . . 118
10.2 Testverfahren . . . 118
10.2.1 Unit-und Integration Tests . . . 118
10.2.2 Laufzeittests . . . 119

10.2.2.1 Testszenario . . . 119
10.2.2.2 Selenium . . . 125
10.2.3 Verfolgungssystem . . . 126
10.3 Testberichte . . . 128
10.3.1 Testbericht Alpha 13. Mai 2010 . . . 128
10.3.2 Testbericht 29. Mai 2010 . . . 133
10.3.3 Testbericht 13. Juni 2010 . . . 138
10.3.4 Testbericht 03. Juli 2010 . . . 143
10.3.5 Testbericht 24. Juli 2010 . . . 148
10.3.6 Testbericht 13. August 2010 . . . 150
10.3.7 Testbericht 28. August 2010 . . . 153
10.3.8 Testbericht 30. September 2010 . . . 156

11 Zusammenfassung und Ausblick 159

12 Fazit der Projektgruppe 161

V Anhang 162

A Installationsanleitung 163
A.1 Voraussetzungen . . . 163
A.2 Storm erweitern . . . 164
A.2.1 Entwicklungsumgebung einrichten . . . 164
A.2.2 Szenario1: Startseite verändern . . . 165
A.2.3 Szenario2: Komponente erstellen . . . 165
A.3 WAR Container erstellen . . . 167
A.4 Storm auf dem Jboss ausführen . . . 167
A.5 STORM-Templatesystem . . . 168
A.5.1 Grundsätzliche Aufteilung . . . 168
A.5.2 Bereich 'components' . . . 169
A.5.3 Bereich 'templates' . . . 169
A.5.4 Bereich 'css' . . . 170
A.5.5 Bereich 'images' . . . 170
A.5.6 Erstellung eines neuen Designs . . . 170

B Benutzerhandbuch 171

C Externe Homepage 172

D Übersicht über die Rechtekonfiguration 173

E Seminararbeiten 177

F Protokolle 178

Abbildungsverzeichnis

2.1 Roadmap Projekt STORM 17
2.2 Arten von Prototypen ([Nie93], S. 94) 18

3.1 Struktur eines Nachhaltigkeitsberichts der Universität Graz 27

5.1 Beispiel-Konfiguration des Rollenkonzepts 32
5.2 Use Case Diagramm Teil 1 34
5.3 Use Case Diagramm Teil 2 35
5.4 Anwendungsfälle im Bereich Schemaeditor 36
5.5 Anwendungsfälle im Bereich Eingabeeditor 39
5.6 Eingabeeditor: Artikelzustände 44
5.7 Anwendungsfälle im Bereich Ausgabe 45
5.8 Anwendungsfälle im Bereich Usability 48
5.9 Tag Cloud Beispiel Web 2.0 [Quelle: [Ang05]] 53
5.10 Trackback am Beispiel einer Pressemeldung ([Ver09]) 53

8.1 Systemarchitektur 61

9.1 ERD Modulkonzept 64
9.2 ERD Rollenkonzept 66
9.3 Anwendungsebenen nach GRI 81
9.4 Sequenzdiagramm „NB anlegen“ 85
9.5 Oberflächenentwurf „Neuen NB anlegen“ 85
9.6 Navigationshierarchie für den Eingabeeditor 87
9.7 Oberflächenentwurf „Artikel Übersicht“ 89
9.8 Oberflächenentwurf „Artikel bearbeiten“ 97
9.9 Sequenzdiagramm „Artikel speichern“ 97
9.10 Sequenzdiagramm „Artikel zum Infokorb hinzufügen“ 106
9.11 Item to Item Collaborative Filtering ([GL03], S. 4) 110

Tabellenverzeichnis

5.1 Übersicht Rechte 31

9.1 Anwendungsfall: „Registrieren“ 70
9.2 Anwendungsfall: „Anmelden“ 70
9.3 Anwendungsfall: „Profil ansehen“ 70
9.4 Anwendungsfall: „Profil ändern“ 71
9.5 Anwendungsfall: „Profil löschen“ 71
9.6 Anwendungsfall: „Startseite betrachten“ 71
9.7 Anwendungsfall: „Meine Followers ansehen“ 72
9.8 Anwendungsfall: „Followers hinzufügen“ 72
9.9 Anwendungsfall: „Follower entfernen“ 72
9.10 Anwendungsfall: „Eigenes Passwort ändern“ 73
9.11 Anwendungsfall: „Passwort zurücksetzen “ 73
9.12 Anwendungsfall: „Protokollierung ansehen“ 73
9.13 Anwendungsfall: „Recht ansehen“ 74
9.14 Anwendungsfall: „Neues Recht hinzufügen“ 74
9.15 Anwendungsfall: „Recht ändern“ 74
9.16 Anwendungsfall: „Recht löschen“ 75
9.17 Anwendungsfall: „Rolle ansehen“ 75
9.18 Anwendungsfall: „Neue Rolle hinzufügen“ 75
9.19 Anwendungsfall: „Rolle ändern“ 76
9.20 Anwendungsfall: „Rolle löschen“ 76
9.21 Anwendungsfall: „Passwort ändern“ 76
9.22 Anwendungsfall: „Benutzer betrachten“ 77
9.23 Anwendungsfall: „Neuen Benutzer anlegen“ 77
9.24 Anwendungsfall: „Benutzer ändern“ 77
9.25 Anwendungsfall: „Benutzer löschen“ 77
9.26 Anwendungsfall: „Konfiguration betrachten“ 78
9.27 Anwendungsfall: „Neue Konfiguration anlegen“ 78
9.28 Anwendungsfall: „Konfiguration löschen“ 78
9.29 Anwendungsfall: „Konfiguration ändern“ 79
9.30 Anwendungsfall: „Modul betrachten“ 79
9.31 Anwendungsfall: „Neues Modul anlegen“ 79
9.32 Anwendungsfall: „Modul löschen“ 80
9.33 Anwendungsfall: „Modul ändern“ 80
9.34 Datenbanktabelle „Report“ 83
9.35 Anwendungsfall „NB anlegen“ 84
9.36 Datenbanktabelle „Article“ 91
9.37 Anwendungsfall „Artikel speichern“ 92
9.38 Anwendungsfall „Datei hochladen“ 94
9.39 Anwendungsfall „Datei runterladen“ 94

9.40 Anwendungsfall „Datei bearbeiten“ 94
9.41 Anwendungsfall „Datei löschen“ 95
9.42 Anwendungsfall „Datei in Artikel einbinden“ 95
9.43 Anwendungsfall „Datei in Ansichten-System einbinden“ 95
9.44 Anwendungsfall „Datei aus Ansichten-System entfernen“ 96
9.45 Anwendungsfall: „Berichte anzeigen“ 98
9.46 Anwendungsfall: „Bericht auswählen“ 99
9.47 Anwendungsfall: „Artikel auswählen“ 99
9.48 Anwendungsfall: „Infokorb anzeigen“ 100
9.49 Anwendungsfall: „Artikel zum Infokorb hinzufügen“ 100
9.50 Anwendungsfall: „PDF/XML aus Artikel erzeugen“ 100
9.51 Anwendungsfall „Artikel kommentieren“ 102
9.52 Anwendungsfall „Artikel bewerten“ 102
9.53 Anwendungsfall „Tags für Artikel vergeben“ 103
9.54 Anwendungsfall: „PDF aus Infokorb erzeugen“ 104
9.55 Anwendungsfall: „XML aus Infokorb erzeugen“ 105
9.56 Anwendungsfall: „RSS-Feed abonnieren“ 105
9.57 Anwendungsfall: „Sprache auswählen“ 114

Teil I.

Einführung

1. Einleitung

Die Projektgruppe (PG) Sustainable Online Reporting Model (STORM) hat während des Wintersemesters 2009/2010 und des Sommersemesters 2010 an der Fakultät II der Carl von Ossietzky Universität ein System entwickelt, mit dessen Hilfe eine Nachhaltigkeitsberichterstattung im WEB 2.0 ermöglicht wird.

In mehreren Workshops, Interviews und Seminaren wurden die fachlichen Anforderungen an eine dialogorientierte Software zur Nachhaltigkeitsberichterstattung ermittelt. Dabei wurde auch die im Wintersemester 2009/2010 parallel laufende Projektgruppe Sustainability Economics and Management (PG SEM) mit einbezogen, deren Ergebnis - eine Analyse der für einen Nachhaltigkeitsbericht wichtigen Indikatoren - in die fachlichen Anforderungen inkludiert wurde. Diese Basis stellte für die konzepionelle Ebene die Basis dar, auf der frühzeitig mit dem technische Entwurf und der Realisierung begonnen werden konnte.

Das Ziel der PG STORM war es, die Idee eines Systems zur Nachhaltigkeitsberichterstattung mit dem Fokus der direkten Integration der Öffentlichkeit respektive der Stakeholder aufzugreifen und mit neuen Technologien und Features, speziell mit den Errungenschaften des WEB 2.0, aufzuwerten und in Bezug auf Dialogorientierung und Usability zu optimieren.

Am Anfang dieses Projektabschlussberichts gilt es zunächst zu erläutern, was im allgemeinen unter einer Projektgruppe verstanden werden kann, bevor auf die Ausgangssituation und Motivation der Projektgruppe STORM eingegangen wird. Anschließend wird auf die Lizenzbedingungen eingegangen. Die Erörterung des Aufbaus dieses gesamten Abschlussberichts bildet den Abschluss der Einleitung und gibt dem geneigten Leser eine Gesamtübersicht.

1.1. Allgemeines zur Projektgruppe

Das Ziel von Projektgruppen ist es, anhand eines gegebenen Problems die vollständige Entwicklung von der Problemanalyse bis zur Realisierung des Systems durchzuführen. Dabei werden Studierende an die berufsnahen Arbeitsweisen wie Teamarbeit, Arbeitsteilung und Übernahme von Verantwortung herangeführt. Desweiteren verstärken die Studierenden ihre persönlichen Fähigkeiten wie das Aufbereiten von Inhalten, das zielorientierte Argumentieren, sowie die Präsentationsfähigkeit. Eine Projektgruppe umfasst in der Regel vier Module, zieht sich über zwei Semester und wird von sechs bis zwölf Studierenden gemeinsam durchgeführt. (Vgl. [Bol06])

An der Carl von Ossietzky Universität Oldenburg werden Projektgruppen in den Fachbereichen Wirtschaftsinformatik, Informatik und Wirtschaftswissenschaften als fester Bestandteil der Master-Studiengänge durchgeführt. Für die Teilnehmer einer Projektgruppe gilt es ein Portfolio an Leistungen zu erbringen. Dieses Portfolio teilt sich in Individual- und Gruppenleistungen. Die aktive und konstruktive Mitarbeit an der Projektgruppe, eine

Seminararbeit und ihre Präsentation sowie die sozialen Interaktionen und der individuelle Einsatz der jeweiligen Person bilden die Einzelnote. Die Zwischenpräsentation, der Abschlussbericht und die -präsentation, der Umfang und die Qualität der Software sind die Kriterien zur Ermittlung der Gruppennote. Nach Beendigung einer jeden Projektgruppe wird diese auf dem Projektgruppentag der Universität Oldenburg vorgestellt. Der Projektgruppentag dient zur Präsentation der Ergebnisse abgeschlossener und der Ideen zukünftigen Projektgruppen. Des weiteren gibt es die Möglichkeit die Ergebnisse der Projektgruppe auf dem Stand des Landes Niedersachsen auf der Messe CeBit vorzustellen. Für die Präsentation der Ergebnisse auf der CeBit hat sich diese Projektgruppe beworben.

1.2. Motivation und Zielsetzung der Projektgruppe

Der Klimawandel, die Reduktion der Ozonschicht und die Verringerung der biologischen Vielfalt sind nur einige ökologische Missstände, die das Ausmaß des menschlichen Eingreifens in die Natur wiederspiegeln. In der Konsequenz ist ein Umdenken auf vielen Ebenen notwendig. (Vgl. [Pfr05], S. 353 ff) „Da die natürlichen Lebensgrundlagen eine notwendige Bedingung für das menschliche Leben und Wirtschaften darstellen, ist ihre Zerstörung inakzeptabel“ ([Rog08], S. 45).

Die Öffentlichkeit wird zunehmend empfindlicher für ökologische Themen, weil auch die Unternehmen zumeist nicht unschuldig an den Umweltschäden sind. Einige Unternehmen veröffentlichen heutzutage bereits Nachhaltigkeitsberichte (Vgl. [Mef92], S. 11). Unter einem Nachhaltigkeitsaspekt in der Öffentlichkeit positiv wahrgenommen zu werden, kann für manche Unternehmen eine Möglichkeit der Differenzierung vom Wettbewerber darstellen. Einige Unternehmen veröffentlichen heutzutage bereits einen Nachhaltigkeitsbericht über das Internet. Dabei werden allerdings auf die Errungenschaften des WEB 2.0 nicht bis wenig zurückgegriffen.

Selbst der Gewinner des Rankings des future e.V. und des Instituts für ökologische Wirtschaftsforschung aus dem Jahr 2009, der Internetbericht von BASF (siehe [fe]) bietet lediglich unter anderem Möglichkeiten zur Weiterempfehlung und Kontaktaufnahme auf Basis von einfachen Formularen. Eine Differenzierung der Nutzer, die Integration der Stakeholder in die Nachhaltigkeitsberichterstattung und die Nutzung der Möglichkeiten des WEB 2.0 sind nicht genutzt. Dabei bietet die internetbasierte Nachhaltigkeitsberichterstattung vielfältige Potentiale: Individuelle Nutzer-Beteiligung, jeden Nutzer als Individuum betrachten, gezielte individuelle Vorschläge, Stakeholder optimal einbinden und somit wertvolles Feedback gewinnen. Dafür ist es notwendig die internetbasierte Nachhaltigkeitsberichterstattung auf Dialogorientierung, Zielgruppenorientierung und Interaktivität auszurichten. Damit einhergehend können weitere Potentiale wie Kostenminimierung und bessere Verbreitungsmöglichkeiten, eine Vertiefungsmöglichkeit der Printberichte und ein digitaler Austausch von Informationen realisiert werden. (Vgl. [Góm08], S. 13 ff u. S. 185 ff)

Ein System zur Veröffentlichung von Nachhaltigkeitsberichten im Internet wurde bereits durch die Projektgruppe „cerebral“ an der Universität Oldenburg geschaffen. Dieses System hatte jedoch die Eigenschaft nur auf statische Möglichkeiten des Web 1.0 ausgerichtet zu sein. Eine dialogbasierte Integration des Anwenders in die Nachhaltigkeitsberichterstattung (NBE) war an dieser Stelle nicht möglich.

Aktuell werden demnach die Stakeholder in die NBE nicht mit einbezogen. Ebenso werden die Möglichkeiten des WEB 2.0 werden wenig bis gar nicht genutzt. Weiterhin hat

die Universität aktuell keine Nachhaltigkeitsberichterstattung. Diese Aspekte und die oben genannten Potentiale stellen die Motivation der Projektgruppe STORM dar. Die Stakeholderkommunikation soll verbessert werden, der Fokus soll auf einer starken Dialogorientierung liegen, zwischen den Nutzern soll differenziert und diesen individuelle Informationen zur Verfügung stellt werden. Kurzum, das Ziel der PG STORM ist die Realisierung einer innovativen, dynamischen, dialogorientierten und integrierenden Web-Anwendung für die Verwaltung, Speicherung und Darstellung von Nachhaltigkeitsberichten.

1.3. Lizenzbedinungen

Bei der Auswahl der Fremdkomponenten wurde darauf geachtet, nur Werkzeuge und Software zu verwenden, die einer Open Source Lizenz unterliegen. Die PG STORM hat sich weiterhin dafür entschlossen, das System STORM unter die General Public Licence (GPL v3) zu erstellen. Diese gewährt dem Autor einen weitestgehenden Schutz von einem Missbrauch eines Produktes. Auch darauf aufbauende Werke müssen ebenfalls unter der GPL veröffentlicht werden. So wird verhindert, dass eine freie Applikation kommerzialisiert wird. (Vgl. [Tho10])

1.4. Aufbau des Projektabschlussberichts

Dieser Projektabschlussbericht besteht auf vier Teilen. Jeder Teil wiederrum enthält mehrere Kapitel. Die Einleitung, an deren Ende sich diese Beschreibung des Aufbaus des Abschlussberichts befindet, stellt das erste Kapitel in der Einführung - dem ersten Teil - dar. Neben dem Teil der Einleitung gibt es als zweiten Teil die Anforderungsdefinition. Der dritte Teil ist der Entwurf. Der Projektabschluss wird im vierten Teil thematisiert wodrauf der Anhang im fünften Teil folgt.
Neben der Einleitung enthält der erste Teil die Kapitel Projektorganisation und Grundlagen zur Nachhaltigkeitsberichterstattung. Hier erfährt der geneigte Leser wie die Projektgruppe organisatorisch und technisch vorgegangen ist und erhälte einen grunlegenden Einblick in die Nachhaltigkeitsberichterstattung. Diese Grundlagen werden in den weiteren Teilen zum ganzheitlichen Verständnis benötigt.
Aufbauend auf den Grundlagen geht es im zweiten Teil um die Anforderungen an ein System zur dialogorientierten Nachhaltigkeitsberichterstattung im Web 2.0. Im Rahmen der Anforderungsdefinition werden zuerst anhand von Vorbemerkungen allgemeine Aspekte beleuchtet. Danach folgen die Erörterungen zu den funktionalen und nicht funktionalen Anforderungen. Hier wird erläutert, welche Funktionen die Software abdecken soll, was also alles möglich sein soll, und welche Eigenschaften und welches Verhalten die Software vorweisen soll.
Auf Basis dieser Anforderungen wird der Entwurf verfasst. Der Entwurf ist Gegenstand vom

2. Projektorganisation

Zur Realisierung einer Webanwendung wie STORM ist ein Projektmanagement und eine Projektorganisation notwendig. In diesem Kapitel erhält der Leser einen Einblick in die Projektorganisation und die Vorgehensweise der PG STORM. Dazu werden zunächst die projektinternen Aufgabenbereiche erläutert. Folgend wird der geplante Projektablauf und das von der PG STORM angewendete Vorgenensmodell erörtert. Weiterhin erhält der Leser ein Überblick über die Softwareunterstützung der Realisierung.

2.1. Aufteilung in Entwicklergruppen

Nachdem sich im Rahmen der Anforderungsanalyse die verschiedenen Anforderungen kategorisieren ließen, beschloss die PG, sich in vier verschiedene Entwicklergruppen aufzuteilen. Hierbei wurde auf die individuellen Wünsche ebenso wie auf die von der PG gesetzte Anforderung, dass jede Entwicklergruppe gemessen an den Anfoderungen adäquat besetzt sein muss, eingegangen. Auf Basis einer demokratischen Einigung ergab sich folgende Gruppenstruktur:

- System Administration (SA): Edzard Fisch, Oliver Norkus
- Web 2.0: Kerem K. Sezer, Christian Wenke, Desislava M. Dechkova, Frank Gerken
- Schema Editor (SE): Swetlana Lipnitskaya, Sebastian van Vliet
- Eingabe Editor (EE): Olaf Roeder, Irene Fröhlich, Gerrit Meerpohl

In der Entwicklergruppe System Administration wurden alle Belange gesammelt, welche die administrative Konfiguration der Software betreffen. So sind in dieser Entwicklergruppe die Bereiche Benutzerverwaltung, Modulverwaltung und das Vorschlagswesen angesiedelt.

In der Gruppe Web 2.0 wurden alle Anforderungen zusammengefasst, die sich aus Errungenschaften des WEB 2.0 direkt und indirekt ergeben. Diese Gruppe kümmerte sich unter anderem um eine zentrale Hilfe, eine Suchfunktionalität und den sogenannten Info-Korb, eine Art digitaler Einkaufswagen für Artikel eines Nachhaltigkeitsberichts.

Das Anlegen eines Schemata für ein NB, sowie das Anlegen, Verwalten und Zuordnen von Indikatoren zu einem NB und ähnliches wurde unter der Bezeichung Schema Editor gruppiert.

Die redaktionelle Befüllung eines Nachhaltigkeitsberichts mit einem Editor, das Gestalten eines Berichts und die Organisation von verwendeten multimedialen Daten in der Mediathek sind Bestandteile der Gruppe Eingabe Editor.

Durch diese Aufteilung in Entwicklergruppen wurde eine fachliche Trennung der Anforderungen in Form dieser Gruppen realisiert, so dass die Entwicklung innerhalb der PG fachlich parallel und in hoher Zusammenarbeit stattfinden konnte.

Um die Entwicklungstätigkeiten zu unterstützen und zu koordinieren wurden projektinterne Aufgabenbereiche geschaffen. Diese werden Folgend erläutert.

2.2. Projektinterne Aufgabenbereiche

In diesem Abschnitt werden die projektinternen Aufgaben, die zu Beginn des Projektes innerhalb der Projektgruppe festgelegt wurden, vorgestellt. Die Rollen selbst wurden auf Basis eines Vorschlags der Betreuer nach einer fachlichen Diskussion ergänzt und von der PG verabschiedet. Innerhalb dieser Diskussion wurde den Rollen ihre Aufgaben zugeteilt und die die jeweiligen Rollen ausfüllenden Personen demokratisch festgelegt.

Hierbei gibt es feste Rollenzuweisungen, die über den gesamten Zeitraum der Projektlaufzeit hinweg gelten. Es wurde gegen eine rollierende Rollenzuweisung entschieden, um den Einarbeitungsaufwand und die Rüstzeiten beim Wechsel einer Rolle zu verhindern. Die jeweiligen Rollen und die Aufgabenbereiche werden im Folgenden vorgestellt, sowie Verantwortlichkeiten genannt.

2.2.1. Projektmanager

Der Projektmanager koordiniert die Projektgruppe im Inneren und ist zentraler Ansprechpartner innerhalb der Gruppe und für die Projektgruppenbetreuung. Der Projektmanager koordiniert und verantwortet die Entwicklung. Er ist Ansprechpartner für die Projektmitglieder bei jeglicher Art von Fragen und Unklarheiten und erste Eskalationsstufe. Der Projektmanager verantwortet die Gestaltung des Projektstrukturplan (PSP). In Abstimmung mit den PG-Mitgliedern wurde der PSP erstellt, diesem wurden pro Entwicklergruppe Aufgaben mit einer Beschreibung und einem Fertigstellungstermin zugeordnet. Der Projektmanager überwacht stetig den Fertigstellungsgrad der Vorgänge und greift koordinierend ein, wenn Abweichungstendenzen zu erkennen sind. Der PSP wurde vom Projektmanager in der Projektmanagement-Komponente der eGroupware (näheres siehe Abschnitt 2.5.1.1) gepflegt. Eine weitere Aufgabe des Projektmanagers ist die Terminverwaltung und -abstimmung. Die Termine und Events der PG werden hierbei in der Kalender-Komponente der eGroupware vom Projektmanager verwaltet. Der Projektmanager ist weiterhin für die Durchführung und Auswertung eines Zeit- und Meilenstein-Controllings verantwortlich. Die Moderation des internen Meetings obliegt dabei ebenso dem Projektmanager. Eine weitere Aufgabe des Projektmanagers ist die Herbeiführung und zielorientierte Diskussion von notwendigen Entscheidungen. Dabei werden alle Entscheidungen demokratisch getroffen. Der Projektmanager ist somit zentraler Ansprechpartner, sowohl für die Projektgruppe als auch für die Gruppenbetreuung und fungiert als Schnittstelle dieser beiden Parteien.
Verantwortlicher: Oliver Norkus

2.2.2. Serveradministrator

Der Serveradministrator hatte die Aufgaben den der PG zur Verfügung gestellten Server zu verwalten. Dazu gehört neben der Installation, Konfiguration und kontinuierlicher Pflege aller benötigten serverseitigen Tools auch die die sichere und stabile Einrichtung des Betreiebssystems, die Wartung und die gesamte Datensicherung. Eine Auflistung und Beschreibung der serverseitig eingesetzen Tools kann im Abschnitt 2.5.1 gefunden werden. Weiterhin war der Serveradministrator der Ansprechpartner für die PG-Mitglieder und Betreuer bei Supportanfragen, wie z.B. beim Zurücksetzen eines Passworts.
Verantwortlicher: Edzard Fisch

2.2.3. Testbeauftragter

Der Testbeauftragte hatte die Aufgabe regelmäßige Blackbox Tests aller Funktionalität der Software STORM auf der Weboberfläche vorzunehmen. Das Verfassen der Testberichte und der Testdokumentation liegt auch in dem Verantwortungsbereich des Testbeauftragten. Ein weiterer Verantwortungsbereich des Testbeauftragten ist die Einrichtung und Pflege der Bugtracking-Komponente der eGroupware (näheres siehe Abschnitt 2.5.1.1) sowie die Initialisierung der Fehlerbearbeitung und die damit zusammenhängende Kontrolle.
Verantwortlicher: Olaf Roeder

2.2.4. Homepage-Beauftragter

Der Homepage-Beauftragte war primär für die initiale Ermittlung der Anforderungen an die Homepage (HP) und deren Realisierung verantwortlich. Die PG hat sich entschieden, kein Content-Management-System zu benutzen, sondern eine eigene Homepage zu programmieren, weil so das STORM-Layout am einfachsten übernommen werden konnte. Die weiteren Aufgaben des HP-Beauftragten waren die Beantragung der Domain und die Verantwortung für die Befüllung der HP mit Content. Die regelmäßige Aktualisierung der HP z.B. in Form der News obliegt ebenfalls dem Homepage-Beauftragten.
Verantwortlicher: Gerrit Meerpohl

2.2.5. Finanzbeauftragter

Der Finanzbeauftragte verwaltete die Projektgruppenkasse (PG-Kasse). Das Guthaben der PG-Kasse ergibt sich aus den Einzahlungen der Beiträgen der Projektgruppenmitglieder. Es wurde entschieden auf monatliche Einzahlungen und Strafeinzahlungen (z.B. bei verspätetem Erscheinen zu einem Meeting) zu verzichten. Die Kasse wurde daher bedarfsweise befüllt. Beträge wurden für die Verpflegung während der Meetings - z.B. Kaffee, Tee - erhoben.
Verantwortlicher: Sebastian van Vliet

2.2.6. Kommunikationsbeauftragter

Die primäre Aufgabe eines Kommunikationsbeauftragten war die Abstimmung mit der PG SEM (siehe hierzu Abschnitt 1). Terminabsprachen, Abstimmung bezüglich des gemeinsam durchgeführten Workshops und Nachbesprechungen waren hierbei Teilaufgaben. Der Kommunikationsbeauftragte koordinierte die Übermittlung von Dokumenten zwischen den Projektgruppen und analysierte in dieser Funktion auch das Ergebnis der PG SEM. Weiterhin obliegt es dem Kommunikationsbeauftragter den Auftritt der beiden Projektgruppen auf einer gemeinsamen Webseite zu gestalten und zu verwalten.
Verantwortlicher: Frank Gerken, Kerem K. Sezer

2.2.7. Layoutbeauftragter

Der Layoutbeauftragte hatte die Hauptverantwortlichkeit für das Layout - technisch gesprochen für CSS/HTML - und für den Klick-Prototyp (siehe auch 2.4). Der Layoutbeauftragte war Ansprechpartner für alle PG-Mitglieder für Layout- oder Gestaltungsfragen.
Verantwortlicher: Frank Gerken

2.2.8. Dokumentationsbeauftragter

Der Dokumentationsbeauftragte war für die Erstellung und Verwaltung aller benötigten Dokumente verantwortlich. Diese sind:

- Vorlage für die Projektdokumentation
- Vorlage für das Protokoll der Gruppentreffen
- Präsentationsvorlagen
- Vorlage für die Seminararbeiten

Verantwortlicher: Christian Wenke

2.3. Projekt Roadmap

Das Projekt STORM war für die Zeit vom 01.09.2009 bis zum 30.09.2010 geplant. Dabei ist diese Zeit in zwei Phasen - entsprechend dem universitären Ablauf von zwei Semestern - eingeteilt.

In der ersten Phase soll eine grundlegende Einarbeitung in das Thema erfolgen. Diese wird anhand einer Seminararbeitsphase realisiert. Die PG-Mitglieder erarbeiteten eine Seminararbeit zu einem Thema, gewählt aus der von den Betreuern vorgegebenen Menge, und präsentierten diese im Rahmen eines PG-Meetings. Auf diese Weise wurden grundlegende Kenntnisse bei den einzelnen Projektmitgliedern aufgebaut (Die angefertigten Seminararbeiten befinden sich im Anhang dieses Abschlussberichts, siehe Anhang E). Weiterhin wurden in der ersten Phase für die Umsetzung des Systems notwendigen Dokumente (Anforderungsdefinition und Entwurf) auf Basis mehrerer Interviews mit den Betreuern dieser PG erstellt.

Am Ende der ersten Phase und vor allem in der zweiten Phase sollte das eigentliche System erstellt werden. Dafür stand dem Projektteam die Zeit von ca. 01.04.2010 bis zum 30.09.2010 zur Verfügung. In diesem zweiten Arbeitsabschnitt sollte das System durch Anwendung einer Vorgehensweise zur schnellen Softwareentwicklung (Rapid Prototyping) erstellt und verschiedene Entwicklungsstände (Prototyp Alpha, Prototyp Beta) präsentiert werden (Näheres hierzu siehe im Abschnitt 2.4). Ein detaillierter Überblick über den Verlauf des Projektes kann der nachstehenden Roadmap (Abbildung 2.1) entnommen werden.

Diese Roadmap verdeutlicht nochmal den Ablauf, aufgeteilt in zwei Semestern, in denen die PG stattfindet. Die einzelnen Meilensteine - in der Roadmap mit einer Raute dargestellt - verdeutlichen die terminliche Vorgehensplanung und geben Aufschluss darüber, wann welcher Meilenstein - in der rot hinterlegten Rechtecke zu finden - erreicht werden soll. Erkennbar ist in der Abbildung auch die parallele Erstellung des Abschlussberichts zur weiteren Entwicklung und Dokumentation.

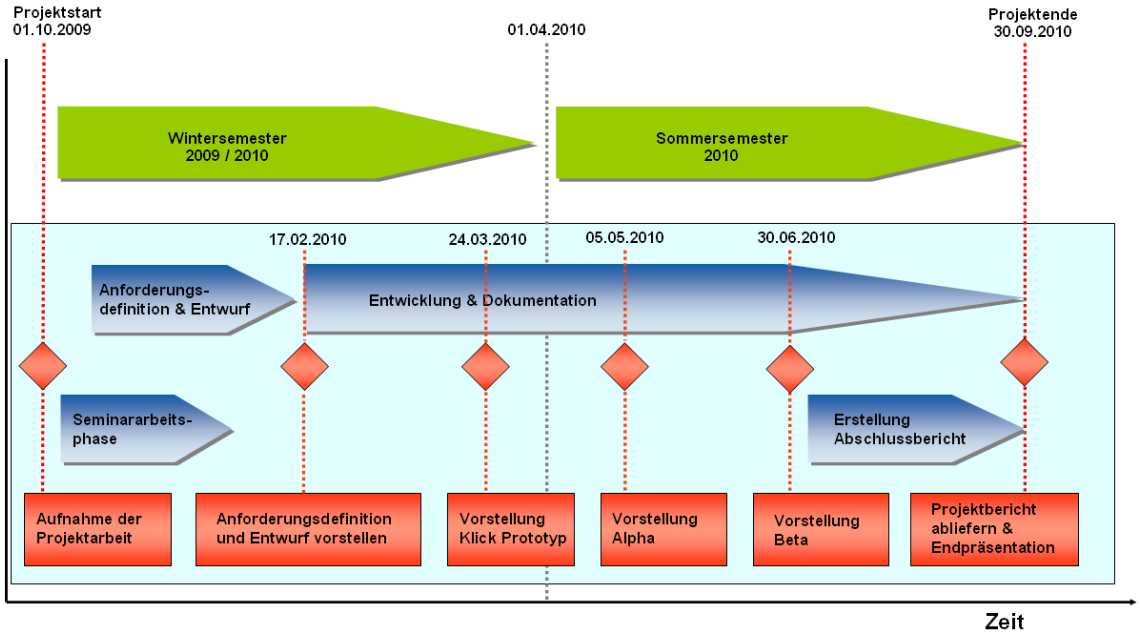

Abbildung 2.1.: Roadmap Projekt STORM

2.4. Vorgehensweise der Projektgruppe

Ein Vorgehensmodell beschreibt die Vorgehensweise der Softwareentwicklung und ist ein ausgefeilter Plan mit zeitlichen und sachlichen Vorgaben. Das Vorgehensmodell legt die Reihenfolge der einzelnen Schritte mit den jeweiligen Ergebnissen fest und regelt, welche Rolle zu welchem Zeitpunkt welche Aktivität in einem Projekt zu erledigen hat. Ein Vorgehensmodell dient zur Steuerung der Softwareentwicklung von der Konzeption bis zum Einsatz. Durch ein Vorgehensmodell wird die Softwareentwicklung übersichtlicher gestaltet und die Beherrschung der Komplexität erleichtert. (Siehe [HH07], S. 170)

Hierbei wird unterschieden zwischen dem methodischen und dem technischen Vorgehen. Das methodische Vorgehen basiert auf dem Vorgehensmodell *Rapid Prototyping*. Das technische Vorgehen beschreibt die Art und Weise des technischen Fortschritts bei der Entwicklung und Auslieferung und basiert auf dem Konzept der *Continuous Integration*.

Hierbei wird unterschieden zwischen dem methodischen Vorgehen, welches auf dem Vorgehensmodell *Rapid Prototyping* basiert, und dem technischen Vorgehen, welches die Art und Weise beschreibt, wie die Auslieferung und Entwicklung technisch fortschritt, und auf dem Konzept der *Continuous Integration* basiert. Diese beiden Aspekte der Vorgehensweise werden folgend erläutert.

2.4.1. Methodische Vorgehensweise: Rapid Prototyping

Das von der Projektgruppe eingesetzte Vorgehensmodell ist das von den universitären Betreuern der Projektgruppe vorgegebene *Rapid Prototyping*. „Unter dem Rapid Prototyping versteht man das schnelle (rapid) Erstellen eines lauffähigen Systems, das wesentliche Eigenschaften des endgültigen Systems besitzt.“ ([For07], S. 186)

Eine zyklische Softwareentwicklung mit Präsentation von Prototypen ist notwenig, da diese Art der Vorgehensweise die schnelle Einbeziehung neuer Erkenntnisse gestattet und wesentlich die Kommunikation zwischen Auftraggebern und Entwicklern unterstützt. (Siehe [For07], S. 186)
Das von der Projektgruppe angewendete Vorgehsmodell wird im Folgenden unter dem praktischen Fokus, wie es in der Realität durchgeführt wurde, beschrieben.

Zunächst wurden die relevanten Dokumente Anforderungsdefinition und Entwurf erstellt. Die Anforderungsdefinition wurde auf Basis von Interviews mit dem Kunde - hier die Betreuer der PG - erstellt. Darauf aufbauend wurde die systemtechnische Realisierung der Anforderungen geplant und im Entwurf beschrieben. An dieser Stelle wurde auf agile Elemente zurück gegriffen. Die Anfoderungsdefinition und der Entwurf konnten so fortlaufend angepasst und konkretisiert werden, so war auch die Aufnahme weiterer Anforderungen jederzeit möglich.

Nachdem die Anforderungen und der Plan zur technsichen Realisierung grundlegend fertig waren, wurde der Startschuss für die Entwicklung gegeben. Fortan wurde dem Kunden in einem zwei-wöchentlichen Rythmus der aktuelle Stand der Software gezeigt und dieser Stand diskutiert. So konnte sich die Software dem *moving target* immer weiter annähern und der Kunde wurde nicht erst in einem späten Stadium mit der Software konfrontiert.

Als erster Auslieferungsstand wurde der Klick-Prototyp gewählt. Der Klick-Prototyp ist ein horizontaler Prototyp, der sich rein auf die Benutzungsoberfläche beschränkt (Siehe ([Nie93], S. 93ff)). In Abbildung 2.2 wird zwischen dem horizontalen und dem vertikalen Prototypen unterschieden. Ein horizontaler Prototyp deckt alle Features grundlegend ab. Ein vertikaler Prototyp erfüllt ein Feature in Gänze.

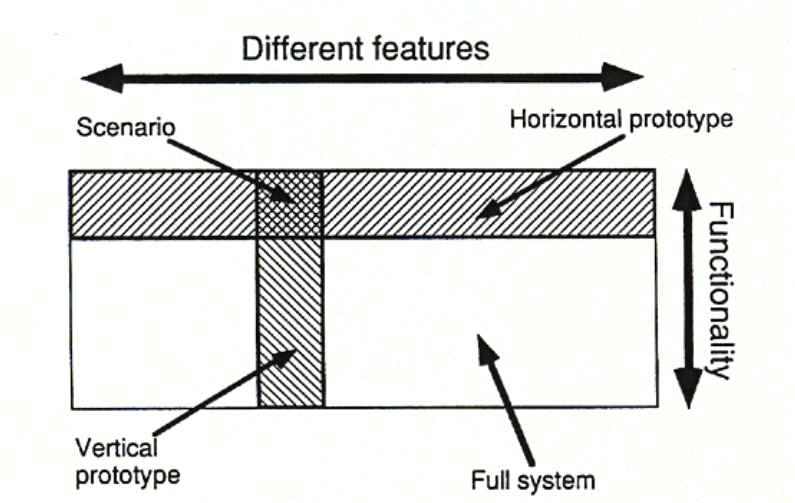

Abbildung 2.2.: Arten von Prototypen ([Nie93], S. 94)

Neben der Vorstellung des Klick-Prototypen wurden zwei weitere Stände ausgeliefert. In der Reihenfolge der Nennung gab es die Alpha- und die Beta-Präsentation. Diese Auslieferungsstände (Releases) werden wir in der softwarebrache üblich mit Buchstaben des griechischen Alphabets bezeichnet. Im Rahmen der Alpha- und Beta-Präsentationen wurde der jeweilige Stand der Software, das Alpha- bzw. das Beta-Release, die Systemarchitektur und die weitere Vorgehensplanung vorgestellt. Da die Beta-Präsentation auch

projekt-fremde Gäste anlockte, wurden auch auf einige Grundlagen (wie sie auch im Kapitel 3 zu finden sind) eingegangen. Durch die Zwischenauslieferungen wurde der Rythmus der inkrementellen Softwarepräsentation vor dem Kunden im zwei-wochen Zyklus nicht beeinflusst.

Durch diese inkrementelle Softwareerstellung konnte die PG gezielt auf die Wünsche und Anforderungen der Kunden eingehen, so war es der PG auch möglich, anfangs noch nicht bekannte und erst im laufenden Prozess entstandene Anforderungen des Kunden zu berücksichtigen. Durch die Abschlusspräsentation, zu der auch Personen aus dem weiten Umfeld der Nachhaltigkeitsforschung eingeladen waren, und die Auslieferung des Finalen-Release wurde das Projekt schließlich beendet.

Um während des einjährigen Projektablaufs alle offenen Punkte zu besprechen, gab es zwei Formen von Meetings, die im wöchentlichen Rythmus durchgeführt wurden: offizielle und interne Meetings.

Im offiziellen Meeting waren neben der PG auch die universitären Betreuer - die Kunden - anwesend. In diesem Meeting wurde - wie oben erwähnt - alle zwei Wochen der aktuelle Stand der Software präsentiert. Es wurden weitere Anforderungen diskutiert und aufgenommen. Dieses Meeting diente zur Koordination und Abstimmung mit dem Kunden. Die Moderation dieses Meetings wurde von der PG mit wechselndem Moderator in alphabetischer Reihenfolge übernommen. Die Protokolle dieser offiziellen Meetings sind im Anhang F zu finden.

Das interne Meeting fand unter Abwesenheit des Kunden statt. Hier wurden projektinterne Diskussionen durchgeführt und über alle Entscheidungen demokratisch abgestimmt. Das interne Meeting diente zur internen Koordination der PG, zur Durchführung des Zeit- und Meilensteincontrollings sowie zur Entwicklergruppen-übergreifenden Abstimmung. Die Moderation des internen Meetings oblag dem Projektmanager (näheres zur dieser Rolle siehe Abschnitt 2.2.1).

2.4.2. Technische Vorgehensweise: Continuous Integration

Dieser Abschnitt soll einen Überblick über die technische Vorgehensweise geben. Zur Realisierung dieser Vorgehensweise wurde nach dem Vorgehen der kontinuierlichen Integration vorgegangen. Kontinuierliche Integration (Continuous Integration, CI)ist ein Begriff, der den Prozess des regelmäßigen, vollständigen Neubildens und Testens einer Anwendung beschreibt. Die grundlegende Idee hierbei ist die fortlaufende Integration, die sich wie folgt niederschlägt: Jeder Entwickler soll frühzeitig und oft Änderungen in die Versionsverwaltung festschreiben. Regelmäßig wird nun das Gesamtsystem neu gebaut (Stichwort: Build) und getestet (Siehe [Fow06]).

Innerhalb der PG wurde Continuous Integration genutzt, um den Entwicklungsstrang zu einem bestimmten Datum - dem sogenannten Change Freeze - zu trennen. Auf dem Hauptzweig der Entwicklung - dem sogenannten Trunk - konnten weiter Funktionalität entwickelt werden. Auf dem Nebenzweig - der Branche - wurden fortan Tests durch den Testbeauftragten realisiert. Im zweiwöchentlichen Rythmus wurde - wie im methodischen Vorgehen beschrieben - dem Kunden der aktuelle Stand der Software präsentiert. Dazu wurde jeweils 5 Tage vorher ein Change Freeze durchgeführt, so dass nach dem damit zusammenhängenden Test und nach Behebung der durch diesen Test aufgedeckten Bugs dem Kunden ein lauffähiges Produkt auf dem Stand der aktuellen Funktionsentwicklung präsentiert werden konnte.

Zu den Auslieferungsständen (Klickprototyp, Alpha und Beta) wurden jeweils mit längerer Vorlaufzeit Change Freezes erstellt und das Produkt ausführlicheren Tests unterzogen. Hierbei galt es auch wieder aufgedeckte Bugs zu beheben, so dass bei der jeweiligen Präsentation ein fehlerfreies, lauffähiges Produkt gezeigt werden konnte. Der finale Change Freeze wurde in Form eines Entwicklungsendes hinsichtlich Funktionalität drei Wochen vor Projektende realisiert. Danach galt es noch Kommentierungen, Fehlerbehebungen und die Internationalisierung der Sprache vorzunehmen. Nach jeder Software-Präsentation wurden die Entwicklungszweige Trunk und Branche wieder zusammengeführt, es fand das sogenannte Mergen statt.

2.5. Softwaretechnische Unterstützung im Projekt

Um ein Softwareentwicklungsprojekt wie STORM durchzuführen, bedarf es einer Reihe von Software zur Unterstützung dieses Projektes und der Entwicklung. Als Grundlage hierzu diente ein Server, welcher von der Abteilung VLBA der Projektgruppe zur Verfügung gestellt wurde. Konkret handelte es sich bei diesem zur Verfügung gestellten Server um einen virtuellen Server - auf einer bestehenden Installation wurde also ein weiteres Betriebssystem virtualisiert.
Ein Server ist ein Rechner, deren zur Verfügung gestellte Dienste von einem Client genutzt werden. Ein virtueller Server ist ein Server, der mit anderen virtuellen Servern auf derselben Hardware ausgeführt wird, für den Benutzer aber als physischer Server sichtbar ist. Das auf diesem Server installierte Betriebssystem ist „Ubuntu“. Auf die Installation dieses Betriebssystems hatte die PG keinen Einfluss, der Serveradministrator konnte lediglich Konfigurationen und weitere Softwareinstallationen vornehmen.

Die von der PG verwendete Software lässt sich anhand von zwei Dimensionen mit jeweils zwei Ausprägungen kategorisieren. Die allgemeinen, wichtigsten Kriterien zur Auswahl der Software-Komponenten waren einerseits die Bedingung, dass das Tool unter einer freien Lizenz stehen sollte - also ohne Kosten eingesetzt werden konnte. Andererseits sollte das entsprechende Tool unter dem Betriebssystem Ubuntu funktionsfähig sein.

Dimensionen

Die erste Dimension ist der Unterstützungszweck. Entweder wird die projektinterne Kommunikation unterstützt - hierbei handelt es sich demnach um 'Organisations-Software' - oder es wird die Entwicklung im weiten Sinne unterstützt, sprich die Erstellung von Programm-Code, die Versionsverwaltung des Softwarecodes und die technischen Mittel zur Veröffentlichung der Software.

Die zweite Dimension ist die Unterscheidung nach dem physikalischen Installationsort. Die entsprechende Softwarekomponente wird entweder auf dem Server oder auf dem Client installiert. Serverseitige Software steht zur Nutzung durch die ganze PG zur Verfügung, hat also globalen Charakter. Clientseitige Software wird auf dem lokalen Rechner des jeweiligen PG-Mitglieds installiert und dient dem Nutzer lokal.

Bei der folgenden Beschreibung der zur Projektbearbeitung genutzten Software wird die Dimension Installationsort herangezogen, da es clientseitig keine Organisations-Software zu installieren galt.

2.5.1. Serverseitige Software

Für die serverseitig installierten Softwarekomponenten war der Serveradministrator verantwortlich (nähere Informationen zur Rolle Serveradministrator siehe Abschnitt 2.2.2). Die serverseitigen Softwarekomponenten können der Dimension *Unterstützungszweck* folgengend weiter unterteilt werden. Auf Basis dieser weiteren Unterteilung werden die einzelnen serverseitigen Softwarekomponenten hier beschrieben.

2.5.1.1. Organisations-Software

Organisations-Software steht hier für Software, die zur projektinternen Kommunikation und Abstimmung genutzt wurde. Hier werden alle Kommunikationssysteme, wie z.B. das Bugtrackingsystem, die Wiki und das Forum zusammengefasst. Im Folgenden werden alle Softwarekomponenten beschrieben, die zur projektinternen Kommunikation und Abstimmung dienten.

eGroupware
Allgemein ist unter Groupware ein computerbasiertes System zu verstehen, welches eine Gruppe von Personen in einem Aufgabengebiet oder Ziel unterstützt und eine Schnittstelle für eine geteilte Arbeitsumgebung bietet ([CAE91] S. 38 ff). Eine Kollaborationssoftware mit dem Ziel der Unterstützung der Zusammenarbeit ist demnach eine Groupware. Das Softwareprodukt 'eGroupware' ist eine Ausprägung einer freien Mehrbenutzer-Groupware. Das Softwareprodukt 'eGroupware' stand als Ergebnis eines Auswahlprozesses dar. Im Rahmen dieses Auswahlprozesses wurden verschiedene Open-Source-Groupware-Produkte (z.B. Open Xchange, Scalix, eGroupware und Zimbra) betrachtet. Die Entscheidung fiel auf die eGroupware, weil dieses Tool aus Sicht der PG alle Anforderungen erfüllt und hierbei auf unnötige Komplexität verzichtet, indem nicht notwendige Komponenten deaktiviert werden können. Die einzelnen Anforderungen werden im folgenden Anhand, der die befriedigenden eGroupware-Komponenten dargestellt.

- Adressbuch/Ressourcenverwaltung: Innerhalb dieser Komponente konnte zu den PG-Mitgliedern Kontaktdaten hinterlegt werden. Die Zugehörigkeit zu einer Entwicklergruppe als wichtige Voraussetzung für ein optimal einsetzbares Bugtrackingssystem konnte gepflegt werden. Erinnerungsmails zur Terminen und Aufgaben konnten automatisch verschickt werden.
- Projektmanagement: Mit dieser Komponente kann ein Projektstrukturplan erstellt werden, Vorgänge angelegt und terminiert werden. Diese Komponente bietet eine Übersicht über die Vorgänge und ermöglicht eine variable Gruppierung der Vorgänge. Somit konnten Aktivitäten und Meilensteine gepflegt und deren Erreichung kontrolliert werden.
- Bugtracking: Hiermit ist eine einfache Verwaltung von Bugs möglich. Ein Bug ist ein in der Software STORM festgestellte Fehler, der zur Bearbeitung durch die einzelnen Entwicklergruppen bereit ist. Kategorisierungen hinsichtlich Entwicklergruppen, eigene Status, Fertigstellungsgrad und Fertigstellungstermin sind mit dieser eGroupware-Komponente realisierbar.
- Wissensmanagement: Mit der eGroupware ist die Realisierung eines Wiki's möglich. Dies plante die PG zur Verwaltung von wichtigen Begriffen, Zusammenhängenden und sonstigen Wissensangelegenheiten.

- Gruppenkalender: Die Anforderung bezüglich eines zentralen Kalenders zur Verwaltung aller Termine und Events erfüllt die eGroupware mit der Komponente Gruppenkalender. Jedes PG-Mitglied konnte einerseits alle Gruppentermine pflegen und einsehen, anderseits aber auch private Termine - wie z.B. Abwesenheiten oder Urlaub - einpflegen. Somit konnte die PG ihre Termine, Events sowie An- und Abwesenheiten zentral verwalten.
- Stundenzettel: Zu Beginn der PG wurde beschlossen, eine Zeiterfassung zu nutzen. Dazu konnte die Komponente Stundenzettel genutzt werden, die es ermöglicht, eine Zeitbuchung direkt einem Vorgang aus dem PSP zuzuordnen.

Die weiteren Komponenten der eGroupware (z.B. die Workflow-Engine) wurden deaktiviert. Dadurch wurde die Komplexität der eGroupware den Anforderungen angepasst.

phpBB
Eine weitere Anforderung der PG an das Kommunikationssystem war die Existenz eines Forums. Ein Forum ist „ein virtueller Platz zum Austausch und Archivierung von Gedanken, Meinungen und Erfahrungen. Die Kommunikation findet dabei asynchron, das heißt nicht in Echtzeit, statt.“ ([Unkc]). Das Forum sollte hier zur Diskussion und zur Abstimmung außerhalb der Meetings genutzt werden, um so auch kurzfristig Entscheigungen herbei zu führen, die nicht bis zum nächsten internen Meeting warten können. Diesbezüglich gab es die Anfoderung an das Forum, eine Umfragefunktionalität zu besitzen. Das in der PG bekannteste Forum, das alle Anforderungen erfüllt, war das Forum 'phpBB'. Somit wurde das php BB Forum in der aktuellen Version 3 benutzt.

Apache2
Um die beiden Kommunikationskomponenten eGroupware und Forum zu Internet zu erreichen, bedarf es eines Web-Servers. „Ein Webserver [...] ist ein Computer, der Dokumente an Clients, wie z.B. Webbrowser überträgt. Als Webserver bezeichnet man den Computer mit Webserver-Software [...].“ ([Unkh]). Um die beiden Kommunikationskomponenten durch einen Webbrowser aufrufbar zu machen, ist eine Webserver-Software notwendig. Als Webserver-Software wurde von der PG die Software 'Apache' selektiert. Der Apache in der Version 2.0 ist ein stabiler und schneller Webserver und erfüllt somit die Anforderung, die beiden Kommunikationskomponenten eGroupware und Forum übers Internet erreichbar zu machen. (Siehe [Foua]).

2.5.1.2. Entwicklungssoftware i.w.S.

Neben der Software zur Unterstützung der Kommunikation und zur Abstimmung innerhalb der PG wurde auch Software zur Unterstützung der Entwicklung im weiten Sinne benutzt. Dazu zählt unter anderem eine serverseitige Versionskontrolle, eine serverseitige Datenbank sowie Software zur Veröffentlichung des aktuellen Standes der Software. Diese Softwarekomponenten zur Unterstützung der Entwicklung i.w.S. werdem im Folgenden dargestellt.

SVN
Apache Subversion (SVN) ist eine Versionsverwaltungssoftware. Dabei erfolgt die Versionierung im zentralen Datenspeicher in Form einer Revisionszählung. Apache Subversion ist die Software zur serverseitigen Versionsverwaltung. Die Clients, die diese Versionsverwaltung nutzen wollen, brauchen eine entsprechende clientseitige Software zum Zugriff auf den zentralen Datenspeicher (Siehe [Foub]). Die Wahl fiel auf Apache Subversion, da

dieses Versionierungstool innerhalb der PG bereits bekannt und von viele Mitgliedern als leistungsfähig eingeschätzt wurde. Ein weiterer Grund waren die positiven Erfahrungen, die der Serveradministrator mit diesem Tool bereits gewonnen hatte.

MySQL
MySQL ist eine Open-Source-Datenbank, welche von der PG aufgrund ihrer guten Leistungsfähigkeit und hohen Zuverlässigkeit gewählt wurde (Siehe [Gmb]). Zudem hatten die meisten PG-Mitglieder mit dieser Datenbank bereits Erfahrungen gesammelt, so dass eine Einarbeitung gering gehalten werden konnte. Auch mit diesem Tool hatte der Serveradministrator bereits positive Erfahrungen erlebt. Damit konnte die PG sich auf in dem PG-Kreis etablierte Tool verlassen und MySQL wurde als serverseitige Datenbank eingesetzt.

Grails
Als Vorgabe der Betreuer galt es, das Web-Applikation-Framework Grails mit der Programmiersprache Groovy zu benutzen. Das Grails-Framework läuft unter einer Java Virtual Machine. Dies hat zur Folge, dass alle weiteren genutzten Komponenten ebenfalls JavaEE-konform sind. (Siehe [Unkd])

JBoss
Der JBoss Application Server ist eine Ausprägung eines JavaEE-Anwendungsservers und ist Teil des JBoss Middleware-Frameworks. JBoss gilt als stabiler Anwendungsserver, der eine hohe Entwicklerproduktivität und Standardkonformität bietet. Ein weiterer Vorteil für die Entscheidung zugunsten JBoss war die gute Integration mit weiteren von der PG genutzten Open-Source-Tools (Siehe [RH]). Über den JBoss Application Server wurde das Softwareprodukt STORM im Web veröffentlicht.

Im Abschnitt Kommunikationssoftware wurde bereits ein Anwendungsserver genannt, der Apache. Demnach werden also zwei Anwendungsserver eingesetzt. Der Grund dafür liegt darin, dass - falls es bei der Entwicklung oder dem CI weitreichende Fehler gäbe und der Anwedungsserver nicht mehr funktionsfähig wäre - auch die digitale Kommunikation nicht mehr möglich wäre. Daher hat die PG entschieden, für die Kommunikationssoftware und für die Entwicklungssoftware i.w.S. zwei verschiedene Anwendungsserver zu betreiben.

Hibernate
Hibernate ist ein JavaEE-basiertes Object-Relational-Mapping-Tool. Object-Relational-Mapping (ORM) ist eine Technik mit der ein in einer objektorientierten Software geschriebenes Anwendungsprogramm, der seine Objekte in einer relationalen Datenbank ablegen kann (Siehe [Unkf]). Hibernate als Software-Tool wird von der selben Organisation herausgegeben und entwickelt, wie der JBoss auch. Insofern eignet sich die Verwendung von Hibernate in diesem Kontext besonders.(Siehe [Hat])
Hibernate dient dazu, zwischen der Datenbank (hier MySQL) und der Entwicklungssprache (hier Groovy) Datenobjekte zu vermitteln. Der Programmierer kann somit durch einfache Befehle ein neues Objekt erstellen und mit diesem beliebig umgehen, ohne sich um die Speicherung dieses Objektes in der Datenbak bemühen zu müssen; dies übernimmt Hibernate durch das ORM.

Hudson
Hudson ist eine Open-Source-Software zur Realisierung einer kontinuierlichen Integration (siehe auch Abschnitt 2.4.2). Mit diesem Tool wurde vom Serveradministrator (genaueres zu dieser Rolle siehe Abschnitt 2.2.2) der Entwicklungsstrag in Trunk und Branche geteilt und wieder zusammengefügt. Dieses Tool diente dazu, die aktuelle Version der Software in den JBoss zu laden und diese Version damit zu veröffentlichen. Hudson ist ein

Java-basiertes CI-Tool, mit dem es einfach möglich ist, Builds zu erstellen, automatisierte Tests durchzuführen und ein ganzheitliches Monitoring über die Entwicklungsstände zu realisieren (siehe [Ora]). Aus diesen Gründen - zuzüglich der positiven Erfahrung des Serveradministrators mit diesem Tool - hat sich die PG für den Einsatz entschieden.

2.5.2. Clientseitige Software

Clientseitige Software ist Software, die auf den persönlichen Rechnern der einzelnen PG-Mitgliedern installiert wird. Diese Softwarekomponenten unterstützen die Softwareentwicklung i.w.S., Kommunikationssoftware wurde rein serverseitig installiert. Um lokal auf den Rechnern der PG-Mitglieder eine Entwicklungsumgebung herzustellen, ist die Installation der hier genannten Software-Komponenten notwendig. Lediglich die Komponente Microsoft Visio unterstützt nicht die Entwicklung, sondern das Erstellen von Diagrammen. Die clientseitigen Softwarekomponenten werden Folgend näher beschrieben.

Microsoft Visio
Ein in der Bedienung einfaches Tool zur Erstellung aller relevanten Dokumente galt es zu finden. Die Wahl fiel auf Microsoft (MS) Visio. Dieses Tool ist nach Erfahrung vieler PG-Mitglieder sehr einfach zu bedienen, deckt die notwenigen Standarddiagramme ab und ist für Studierende über MSDNAA kostenfrei zu bekommen. „MSDN Academic Alliance ist eine Initiative von Microsoft, welche an Hochschulen den Dozierenden sowie Studierenden von Ingenieur- oder Informatikwissenschaften den Zugang zu Microsoft-Technologien eröffnet“ ([Micb]). Auf Basis von vorgefertigten Shapes, Beispielzeichnungen und Vorlagen ist es mit MS Visio einfach Diagramme für die IT-, Unternehmens- und Prozessverwaltung und viele andere Bereiche zu erstellen - diese Shapes können mit der Maus einfach und präzise angeordnet werden, so ist es möglich komplexe Diagramme in kurzer Zeit zu gestalten (siehe [Mica]). Diese positiven Erfahrungen haben einige PG-Mitglieder eingebracht und letztlich hat die PG den Einsatz dieses Tools beschlossen.

IDE
Eine integrierte Entwicklungsumgebung (Abkürzung IDE, von engl. integrated development environment) ist ein Anwendungsprogramm zur Entwicklung von Software (siehe [Unke]). Diese Software wird lokal auf den Entwicklungsrechnern installiert und dient zur Programmierung. Auf der offiziellen Grails-Seite „IDE Integration“ sind verschiedene IDE genannt, die sich zur Entwicklung mit dem Grails-Framework eignen ([Spr]). Diese sind ([Spr]):

- STS
- NetBeans
- IDEA
- TextMate Bundle
- Gedit
- jEdit
- Emacs

Um den Entwicklern eine größtmögliche Freiheit in der Wahl ihrer IDE zu überlassen, hat die PG beschlossen, eine IDE aus der oben genannten Liste zu selektieren. Somit ist sichergestellt, dass von der PG nur Entwicklungsumgebungen benutzt werden, die entsprechend der offiziellen Grails-Seite genannt und somit empfohlen sind.

mySQL
Um lokal auf den Entwicklungsrechnern die Software zu starten und zu testen, musste eine entsprechende Umgebung eingerichtet werden. Zu dieser Umgebung zählt auch die Datenbanksoftware MySQL, welche auch serverseitig zum Einsatz kommt. So wurde sichergestellt, dass client- und serverseitig bezüglich der Datenbank Symmetrie besteht.

SVN-Client
Um auf die serverseitig in der Versionsverwaltung gepflegten Daten zugreifen zu können, ist es notwendig, clientseitig ein entsprechendes Software-'Gegenstück' zu installieren. Hier entschied die PG, dem Entwickler bei der Toolwahl freie Hand zu lassen. Beispielsweise wurde von vielen PG-Mitgliedern das Tool TortoiseSVN eingesetzt. TortoiseSVN ist eine leicht bedienbare Source Control Software für Microsoft Windows. Durch die Integration des Source-Ordners - des Ordners, in dem die Dateien aus der serverseitigen Versionsverwaltung liegen - in den Windows Explorer wird ein optimales Arbeiten ermöglicht. (Siehe [tea])

3. Grundlagen zur Nachhaltigkeitsberichterstattung

Um einen Überblick über die Bedeutung von Nachhaltigkeit und die Erstellung von Nachhaltigkeitsberichten zu erhalten, sollen nachfolgend einige wichtige Grundlagen vermittelt werden. Diese Grundlagen sind zum Verständnis des Projektes von hoher Bedeutung.

3.1. Was sind Nachhaltigkeitsberichte?

Unter „Nachhaltigkeit“ oder „Nachhaltige Entwicklung“ wird der verantwortungsvolle, beständig erhaltende und über die Gegenwart hinaus geplante Umgang mit Ressourcen verstanden. Die drei Bereiche, die dabei innerhalb der Nachhaltigkeitsberichterstattung von besonderer Bedeutung sind, sind Ökonomie, Ökologie und Soziales. Ein Nachhaltigkeitsbericht ist dabei insgesamt ein Mittel um die Stakeholder einer Organisation über diese integralen Bereiche zu informieren und auf dem Laufenden zu halten.

Historisch gesehen handelt es sich bei den Nachhaltigkeitsberichten um die Nachfolger der Umweltberichte, die um ökonomische und soziale Aspekte erweitert wurden. Sie werden dabei parallel zu den notwendigen Geschäftsberichten erstellt, wobei teilweise auch eine Integration eines Geschäftsberichts in den NB erfolgen kann. Bei der Erstellung eines Nachhaltigkeitsberichts ist es üblich geworden, diese an den Richtlinien der Global Reporting Initiative (GRI) zu orientieren. Diese Richtlinien stellen eine Strukturreferenz im Bereich der Nachhaltigkeitsberichterstattung dar und sollen auch innerhalb der Umsetzung des Projektes STORM als Grundlage dienen.

3.2. Internetbasierte Nachhaltigkeitsberichterstattung

Bei Vergleich der aktuellen Entwicklung innerhalb der NBE, so fällt auf, dass zum größten Teil eine statische Bereitstellung der Daten erfolgte und teilweise noch erfolgt. Eine Integration des Benutzers in die Zusammenstellung oder Bewertung der NBs ist an dieser Stelle ebenso wenig möglich, wie der Vergleich von verschiedenen Nachhaltigkeitsberichten. Um hier weitere Interaktionsmöglichkeiten anzubieten, soll das System verschiedene Aspekte des Web 2.0 anbieten. Dies soll vor allem dazu dienen, die Stakeholder in die NBE einzubinden. Unter Stakeholdern verstehen sich in diesem Zusammenhang alle Interessengruppen, an die ein NB adressiert ist. Stakeholder können sowohl die breite Öffentlichkeit, als auch Investoren, Kunden oder Mitarbeiter einer Organisation sein.

Dem Benutzer soll ermöglicht werden Kommentare zu Nachhaltigkeitsberichten abzugeben oder Nachhaltigkeitsberichten zu bewerten. Damit kann ein Aspekt des Web 2.0, das Collaborative Filtering, bei dem die Benutzer auf bestimmte Artikel hingewiesen werden, die für andere Benutzer von Relevanz waren, realisiert werden.

3.3. Aufbau von Nachhaltigkeitsberichten

Das Inhaltsverzeichnis des NBs der Universität Graz gezeigt in Abbildung 3.1 zeigt einen möglichen Aufbau eines Nachhaltigkeitsberichts in der Anwendung an einer Universität. Dies ist für das Projekt relevant, da das Primärziel der Projektgruppe, ist eine Plattform zu schaffen, die zur Erstellung von Nachhaltigkeitsberichten an Universitäten dient.

VORWORT
1 NACHHALTIGE ENTWICKLUNG UND UNIVERSITÄTEN
DIE GEGENWÄRTIGE GLOBALE SITUATION
DIE ROLLE DER UNIVERSITÄTEN IN DER NACHHALTIGEN ENTWICKLUNG
2 DIE UNIVERSITÄT GRAZ ZWISCHEN TRADITION UND INNOVATION
ORGANISATION DER UNIVERSITÄT
DIE ROLLE DER TASK-FORCE NACHHALTIGE UNIVERSITÄT GRAZ
3 LEITBILD DER UNIVERSITÄT GRAZ
MOTTO UND WERTE
MISSION UND ZIELE FÜR LEHRE UND STUDIUM
MISSION UND ZIELE FÜR FORSCHUNG UND WISSENSTRANSFER
MISSION UND ZIELE DER INTERNATIONALISIERUNG
MISSION UND ZIELE FÜR MITARBEITER/INNEN
MISSION UND ZIELE DER GLEICHSTELLUNG, GLEICHBEHANDLUNG UND FRAUENFÖRDERUNG
MISSION UND ZIELE FÜR DEN UMWELTSCHUTZ
4 STUDIUM UND LEHRE
CURRICULUM STRUKTUR UND INHALTE
PERSPEKTIVEN IN LEHRE UND STUDIUM
5 FORSCHUNG
FORSCHUNGSAUSRICHTUNG DER UNIVERSITÄT GRAZ
FORSCHUNGSEVALUIERUNG UND UMSETZUNG
FÖRDERUNG VON NACHWUCHSFORSCHER/INNE/N
INTERNATIONALISIERUNG DER FORSCHUNG
PERSPEKTIVEN FÜR DIE FORSCHUNG
6 SERVICEEINRICHTUNGEN DER UNIVERSITÄT
DIE ÖSTERREICHISCHEN HOCHSCHÜLERINNENSCHAFT (ÖH) AN DER UNIVERSITÄT GRAZ
„4STUDENTS" INFO SERVICE
DIE UNIVERSITÄTSBIBLIOTHEK
ZENTRALER INFORMATIKDIENST
UNIVERSITÄTSSPORTINSTITUT
BÜRO FÜR INTERNATIONALE BEZIEHUNGEN
7 GLEICHSTELLUNG, GLEICHBEHANDLUNG UND FRAUENFÖRDERUNG
FRAUENFÖRDERUNGSPLAN
BUDGETÄRES ANREIZSYSTEM
KARRIEREPROGRAMM FÜR WISSENSCHAFTLERINNEN
PERSPEKTIVEN
8 SOZIALES UND UMWELTSCHUTZ
BEREICH SOZIALES
BEREICH UMWELTSCHUTZ
9 LEBENSPARTNERIN UNIVERSITÄT
LEBENSPARTNERIN FÜR WEITERBILDUNG UND UMSETZUNG VON FORSCHUNGSERGEBNISSEN
ZENTRUM FÜR WEITERBILDUNG
ALUMNI
SPONSORING UND FUNDRAISING
10 PERSPEKTIVEN FÜR EIN NACHHALTIGKEITSPROGRAMM
11 ZUR BEGLEITUNG DES NACHHALTIGKEITSBERICHTS
12 GRI -RICHTLINIEN

Abbildung 3.1.: Struktur eines Nachhaltigkeitsberichts der Universität Graz

Teil II.

Anforderungsdefinition

4. Vorbemerkungen zur Anforderungsdefinition

In diesem Teil II des Abschlussberichts wird die Anforderungsdefinition beschrieben. Eine Anforderungsdefinition zielt auf eine verbindliche Festlegung von Anforderungen, die an ein zu entwickelndes System gestellt werden. Der Begriff Anforderungsdefinition beschreibt zielgerichtete Tätigkeiten um eine möglichst vollständige und für alle Beteiligten verbindliche Sammlung von Anforderungen zu formulieren, die ein technisches Informationssystem erfüllen soll. (Siehe [Myr].) Die Anforderungsdefinition in diesem Abschlussbericht stellt ein Vertrag zwischen der Projektgruppe und den Betreuern der PG dar. Die Mitglieder der PG haben die Anforderungen im Rahmen von Meetings, Interviews und Workshops ermittelt. Die Anforderungsdefinition ist aufgeteilt in funktionale und nicht funktionale Anfoderungen. Funktionale Anforderungen legen fest, was das Produkt tun soll. Nicht funktionale Anforderung legen fest, welche Eigenschaften das Produkt haben soll. (Siehe [SR06], S. 9f.) Im Folgenden werden zuerst die funktionalen und anschließend die nicht funktionalen Anforderungen an die Software STORM erörtert.

5. Funktionale Anforderungen

Im Rahmen des Web 2.0 wird die rein informative Natur einer Webseite erweitert durch interaktive und kommunikative Aspekte wie zum Beispiel:

- Inhaltliche Vergleichbarkeit von Nachhaltigkeitsberichten
- Verlinkbarkeit von Webseiten untereinander mittels Web 2.0-Technologien wie z.B. Trackback (siehe Abschnitt 5.7.9)
- Collaborative Filtering
- Autovervollständigung
- User generated content[1]

Für die PG STORM werden genau die Funktionen des Web 2.0 interessant, die für eine Erstellung, Zusammenstellung, Kombinierung, Bearbeitung und Vergleichbarkeit von Nachhaltigkeitsberichten verwendet werden können. Die dafür notwendigen funktionalen Anforderungen werden nachfolgend beschrieben.
Begonnen wird hier mit der Erläuterung der Anforderungen an den Bereich der Systemadministration. Es folgt die Beschreibung der Anforderungen an die Bereiche Schemaeditor, Eingabeeditor, an die Ausgabe sowie an Usability-Funktionen und Personalisierung. Mit weiteren Anforderungen, die in die gerade genannten Bereicht nicht gruppiert werden können, endet die Erläuterung der funktionalen Anforderungen.

5.1. Systemadministration

In diesem Kapitel wird auf die Anforderungen bezüglich der Systemadministration eingegangen. Es werden die Anforderungen an das Rollenkonzept erläutert - Flexibilität steht hierbei im Fokus der Anforderung. Anschließend wird eine mögliche Konfiguration des Rollenkonzeptes beschrieben, welche auch als Auslieferungszustand der Software STORM dienen soll. Hiernach werden die Anwendungsfälle aus dem Umfeld der Benutzer- und Rechteverwaltung erörtert. Es folgt die Erläuterung der Anforderungen an den Registrierungsprozess und die Beschreibung des Umgangs mit Passwörtern. Folgend werden die Anforderungen an eine Protokollierung im Bereich Systemadministration genannt. Dieses Kapitel abschließend wird auf die Anforderung der An- und Abschaltbarkeit von Web 2.0-Features[2] eingegangen.

5.1.1. Rollenkonzept

Eine grundlegende Anforderung an die Software ist eine flexibele und erweiterbare Rechte- und Benutzerverwaltung. Jede Person, die mit dem System arbeitet, ist zunächst als anonymer Benutzer für das System sichtbar. Bei der Registrierung des Benutzers wählt dieser

[1] Nutzergenerierte Inhalte (Beispiel: Wikipedia, Youtube, usw.)
[2] Web 2.0-Features sind im weiten Sinne Elemente aus dem Bereich Web 2.0 wie z.B. Social Booksmarks, Kommentierung von Artikeln und die Bewertung von Artikeln.

einen Benutzername aus und vergibt ein Passwort. Dadurch wird er zu einem registrierten Benutzer, der bei der Anmeldung im System authentifiziert wird. Einem Benutzer können Berechtigungen innerhalb des Systems zugeordnet werden. Diese Berechtigungen werden wiederrum durch Benutzer des Systems mit entsprechenden Rechten (Administratoren) aus einer Menge von definierten Rollen gewählt und dem Benutzer zugewiesen.

Über ein Recht wird definiert, ob ein Benutzer der Software einen Bereich nutzen darf oder nicht. Eine beliebige Bündelung von Rechten soll zu einer Rolle zusammengefasst werden können. Diese Rolle soll beliebig vielen Benutzern zugeordnet werden können. Auf diese Weise können die Strukturen und Verantwortlichkeiten innerhalb einer Organisation auch in der Software abgebildet werden.
Der Auslieferungszustand der Software soll über eine Grundmenge an Rechten und Rollen verfügen. Eine Erweiterung dieser Rechte und Rollen wird nur dann realisiert, wenn eine Erweiterungsentwicklung vorgenommen wird.

Die nachfolgende Tabelle 5.1 listet eine Beispielmenge identifizierter Rechte auf. Diese Liste verschafft lediglich einen Überblick und erhebt an dieser Stelle kein Anspruch auf Vollständigkeit. Im Laufe der Entwicklung wird der Prozess der Rechtedefinition weiter aktiv sein - zu jeder neuen Seite wird ein neues Recht definiert werden. Diese Liste dient hier nur zur Veranschulichung des Rollenkonzeptes.

Rechte
Registrieren
Profil verwalten
Artikel bewerten
Artikel kommentieren
Artikel bearbeiten
Erstellen von Schemata
Ändern von Schemata
Ändern von Schemata
Rollen verwalten

Tabelle 5.1.: Übersicht Rechte

Zur Förderung der Verständlichkeit werden hier zwei Rechte näher erläutert. Eine ausgearbeitete Beispielkonfiguration, welche auch die Auslieferungskonfiguration der Software darstellt, ist in Abschnitt 5.1.2 gezeigt.
Das Recht *Erstellen von Artikel im NB* erlaubt das Hinzufügen von neuen Artikeln in einem NB. Beispielsweise könnte es eine Rolle Redakteur geben, welche dieses Recht besitzt. Ein Benutzer, dem gegebenenfalls neben anderen Rollen auch die Rolle Redakteur zugeordnet ist, dürfte einem NB neue Artikel hinzufügen.
Das Recht *Rollen verwalten* gestattet es, einem Benutzer mit diesem Recht, beispielsweise einem Administrator, vorhandenen Rollen Benutzer und Rechte zuzuordnen.

5.1.2. Beispiel-Konfiguration des Rollenkonzepts

Eine mögliche Beispiel-Konfiguration des Rollenkonzepts ist in Abbilung 5.1 dargestellt. Diese Abbildung ist in drei Bereiche eingeteilt: User (Benutzer), Rollen und Rechte. Deutlich wird an dieser Abbildung, dass die Benutzer ihre Rechte über die Zuordnung zu einer Rolle bekommen. Die in der Abbildung blau dargestellte Verbindung verknüpft Benutzer und Rollen. Die rote Verbindung ordnet Rechte und Rollen einander zu. Durch die zweistufige, variable Zuordnung wird maximale Flexibilität erreicht.
Der *User4* hat beispielsweise über die Rolle *Registrierter* die Rechte *Profil verwalten*, *Artikel bewerten* und *Artikel kommentieren*. Daneben hat der *User4* über die Rolle *Redakteur* die Rechte *Artikel bearbeiten*, *Erstellen von Schemata* und *Ändern von Schemata*. In seiner Rolle *Admin* darf der *User4* die Rollen auch verwalten und beliebig neu zuordnen.

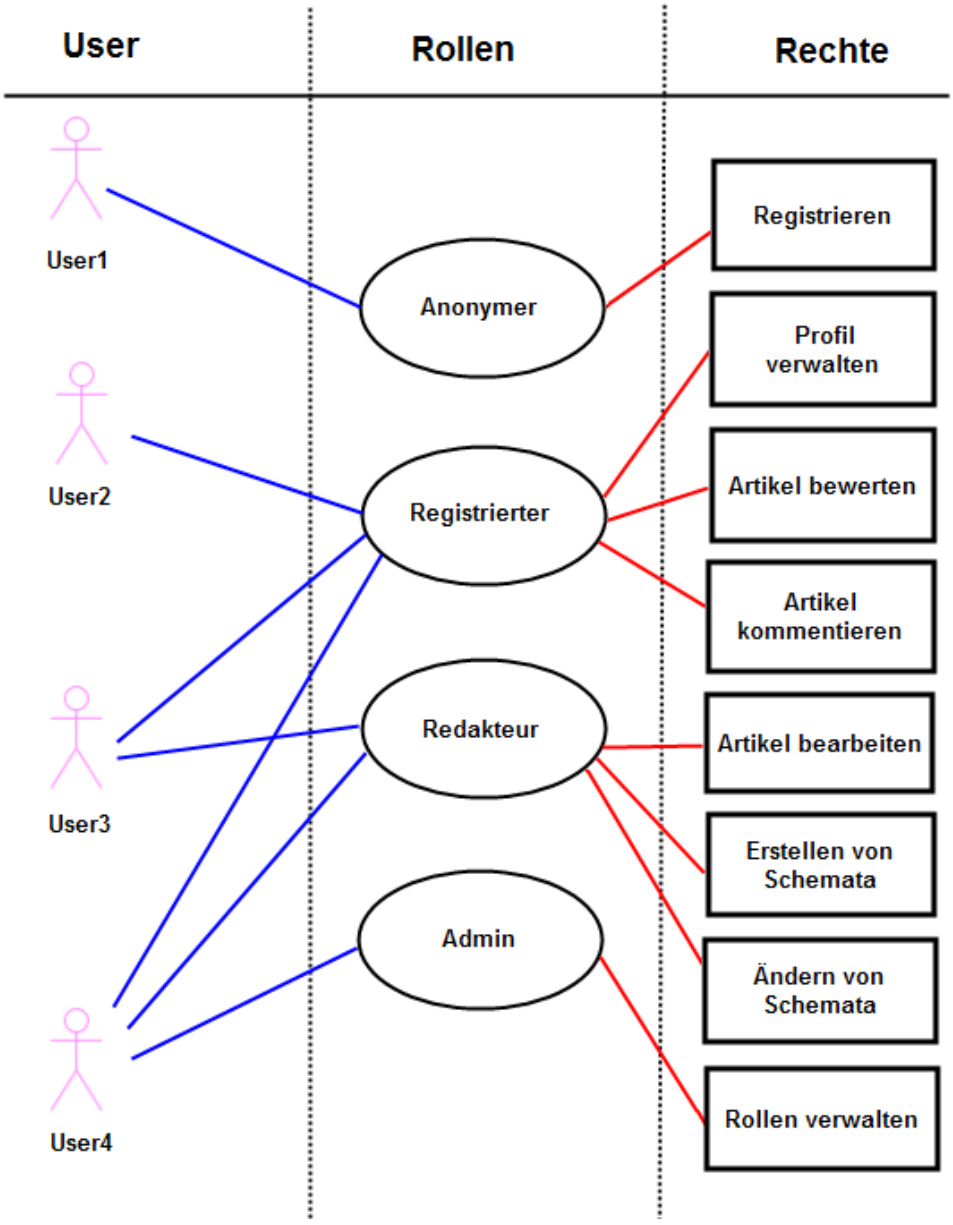

Abbildung 5.1.: Beispiel-Konfiguration des Rollenkonzepts

Diese Beispiel-Konfiguration soll auch die Konfiguration des Rollenkonzepts im Auslieferungsstand der Software STORM darstellen. Der Auslieferungszustand dient explizit zur Adaption an spezifische Belange.

5.1.3. Anwendungsfälle im Bereich Benutzer- und Rechteverwaltung

Das nachfolgende Anwendungsfalldiagramm, hier wegen der Übersichtlichkeit aufgeteilt in Abbildung 5.2 und 5.3, veranschaulicht alle Anwendungsfälle des Bereichs Systemadministration. Dabei ist jeweils die Rolle aufgeführt, die dem Benutzer zugeordnet sein muss, um den jeweiligen Anwendungsfall durchführen zu können.

5.1.4. Registrierungsprozess

Im Rahmen der Registrierung eines neuen Benutzers, soll eine Registrierungs-Bestätigungs-E-Mail verschickt werden. Der Benutzer tippt seine Daten in das Registrierungsformular ein. Sobald er dies absendet, soll eine Registrierungs-Bestätigungs-E-Mail an die vom Benutzer eingetragene E-Mail-Adresse verschickt werden, welche einen Registrierungs-Bestätigungs-Link enthält. Erst nach dem Klick auf diesen Link soll der Benutzer für die Anmeldung am System freigegeben sein. Somit soll die Prüfung einer korrekten, verwendeten E-Mail-Adresse realisiert werden. Dies ist auch für den Anwendungfall Passwort zurücksetzen notwendig.

5.1.5. Umgang mit Passwörtern

Den Benutzern soll es möglich sein, ihr Passwort nach positiver Anmeldung selbstständig zu ändern. Weiterhin sollen Benutzer, die ihr Passwort vergessen haben, dies eigenständig zurücksetzen können. Hier soll ein Self-Service realisiert werden, der dem Benutzer sein neues Passwort per E-Mail zusendet.

5.1.6. Protokollierung

Aus Sicherheitsgründen und zur Nachvollziehbarkeit sollen die An- und Abmeldungen am System protokolliert werden.

5.1.7. An- und Abschaltbarkeit

Es gibt die Anforderung der An- und Abschaltbarkeit von Web 2.0-Features. Die Features, wie z.B. Social Booksmarks, Kommentierung von Artikeln und die Bewertung von Artikeln, sollen vom Administrator der Software beliebig an- und abschaltbar sein. Dadurch wird Organisationen, die diese Software verwenden, die Möglichkeit geben eigenständig zu entscheiden, ob sie ihre Berichtsinhalte z.B. öffentlich kommentieren lassen möchten oder nicht. Nach erfolgreicher Abschaltung soll dieses Feature nicht mehr auf der entsprechenden Seite zu sehen sein. Beispielsweise soll nach dem Abschalten des Features „Kommentierung von Artikeln“ unter einem Artikel das Kommentarfeld nicht mehr auftauchen. Nach erfolgtem Anschlaten soll dieses Feld wieder sichtbar sein.

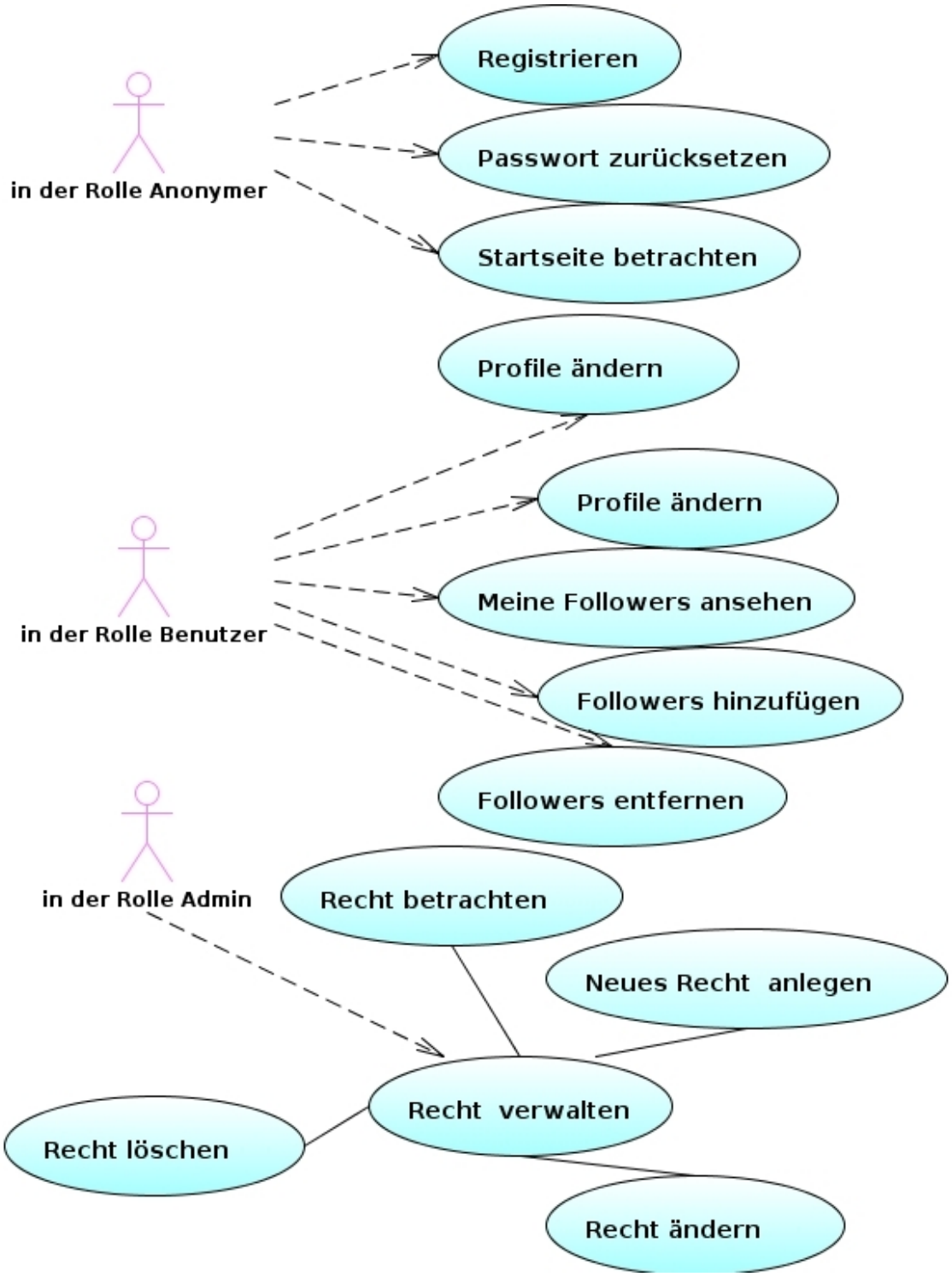

Abbildung 5.2.: Use Case Diagramm Teil 1

Abbildung 5.3.: Use Case Diagramm Teil 2

5.2. Schemaeditor

Ein Schemaeditor soll zur Erstellung und Verwaltung eines Schemas dienen, auf dessen Grundlage das Grundgerüst eines NBs automatisch generiert werden kann. Eine grundlegende Anforderung an den Schemaeditor ist es, den GRI-Berichtsrahmen als Vorgabe für NBs abbilden zu können (siehe Kapitel **??**. Dadurch soll ebenfalls die Validierung eines NBs gegen das Schema, auf welchem dieser basiert, ermöglicht werden. Für die Erstellung, Verwaltung und Freigabe des Schemas wird im Sinne des definierten Rollenkonzeptes (siehe Kapitel 5.1.1) ein Benutzer des Schemaeditors als „Schemaverwalter“ definiert. Die verschieden Anwendungsfälle im Bereich Schemaeditor zeigt die Abbildung 5.4.

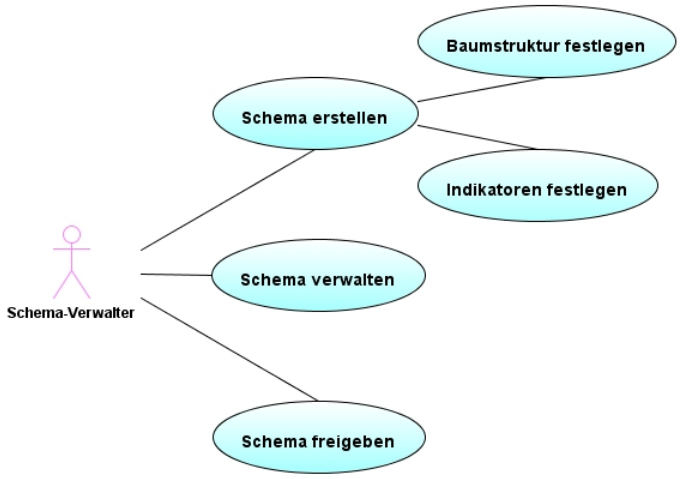

Abbildung 5.4.: Anwendungsfälle im Bereich Schemaeditor

5.2.1. Das Schema-Modell

Für eine erfolgreiche Validierung eines NBs, müssen innerhalb eines Schemas eine Struktur und Inhaltsvorgaben -auch Indikatoren genannt- definiert werden, nach denen ein NB geprüft werden kann. Hierbei soll jedoch die Struktur eines Schemas nicht als Voraussetzung, sondern als ein Vorschlag für einen NB dienen, während die Inhaltsvorgaben zwingend in einem NB vorhanden sein müssen. Die Struktur eines Schemas wird aus verschiedenen Ebenen zusammengesetzt, wobei das Hinzufügen neuer Ebenen möglich ist. Diesen Ebenen werden Indikatoren zugeordnet, die in einem späteren NB, welcher das Schema als Grundlage nutzt, übernommen werden müssen.

5.2.2. Schema verwalten

Im Folgenden werden die Anforderungen an den Schemaeditor erläutert.

Schema anlegen

Es kann ein neues Schema angelegt werden, welches persistent in einer Datenbank gespeichert wird und zugreifbar ist.

Schema bearbeiten

Desweiteren kann ein bereits angelegtes Schema modifiziert und erweitert werden. Dabei kann sowohl die Struktur als auch die Indikatorenzuordnung angepasst werden.

Schema löschen

Wird ein vorhandenes Schema nicht mehr benötigt, kann dieses gelöscht werden. Damit wird das Schema sowohl aus der Ansicht als auch aus der Datenbank entfernt.

5.2.3. Indikatoren verwalten

Indikatoren sind eine grundlegende Voraussetzung für ein Schema und nachfolgend für einen NB. Im Folgenden werden die Anforderungen für Indikatoren dargelegt.

Indikator anlegen

Es kann ein neuer Indikator anlegt werden, welcher in einer Datenbank persistent gespeichert wird.

Indikator bearbeiten

Bereits bestehende Indikatoren können bearbeitet werden. Falls der zu bearbeitende Indikator bereits in einem NB referenziert wird, wird die Änderung ebenfalls übernommen.

Indikator löschen

Indikatoren können gelöscht werden, sofern auf diese in keinem NB referenziert wird.

5.2.4. Kategorien verwalten

Indikatoren mit Gemeinsamkeiten können in eine Kategorie zusammengefasst werden zur besseren Übersicht und Bedienbarkeit.

Kategorie anlegen

Es ist möglich, eine neue Kategorie persistent zu speichern, welcher Indikatoren zugeordnet werden können.

Kategorie bearbeiten

Ebenfalls können Kategorien bearbeitet werden. Dabei soll es optional die Möglichkeit geben, Indikatoren dieser Kategorie in eine andere zu verschieben.

Kategorie löschen

Eine Kategorie kann gelöscht werden, sofern dieser keine Indikatoren zugeordnet sind.

5.2.5. Optionale Anforderungen an den Schemaeditor

Neben den zuvor genannten obligatorischen Anforderungen sind die Folgenden optional:

XML-Import/ -Export

Ein Export als auch ein Import von Schemata ist optional möglich. Dies bedeutet, dass ein Schema auf der einen Seite z.B. als XML-File exportiert, gespeichert und ausgetauscht werden kann. Auf der anderen Seite kann ein Schema z.B. als XML-File importiert und benutzt werden.

Schema importieren

Eine weitere optionale Anforderung ist, das Importieren bereits bestehender Schemata in ein gewähltes Schema. Hierbei wird das Schema mit den dazugehörigen Ebenen in das ausgewählte integriert. Falls das importierte Ergebnis nicht zufriedenstellend sein sollte, wird eine Undo-Funktion (Rückgängig) angeboten, zur Wiederherstellung des alten Zustandes.

Ebene verschieben

Desweiteren soll es möglich sein, innerhalb der Struktur eines Schemas, die Ebenen zu verschieben und vertauschen.

Indikatorenverlauf

Wird der gleiche Indikator in mindestens zwei NBs verwendet, können Vergleichswerte in Form verschiedener Diagramme dargestellt werden.

5.3. Eingabeeditor

Der Eingabeeditor ist für die Verwaltung von kompletten Nachhaltigkeitsberichten sowie einzelnen Artikeln zuständig. Inhalte können entweder vom Benutzer über den Eingabeeditor eingegeben werden oder aus externen Datenquellen (z.B. SAP-Systeme) extrahiert werden können. Auf die Funktionen des Eingabeeditors wird im Folgenden näher eingegangen.

5.3.1. Nachhaltigkeitsberichte verwalten

Ein Nachhaltigkeitsbericht kann angelegt werden, dazu ist die Auswahl eines Schemas notwendig. Um für einen beliebigen NB zu vermerken, ob dieser gerade neu, noch bearbeitet werden darf, veröffentlicht ist oder bereits geschlossen wurde, kann sich ein NB in den folgenden Zuständen befinden.

Neu

Nach dem Anlegen eines neuen NBs erhält dieser den Zustand „Neu“ und kann von nun an beliebig bearbeitet werden.

In Bearbeitung

Einem NB im Zustand „in Bearbeitung“ können Artikel hinzugefügt und verwaltet werden. Der NB kann anhand des zugehörigen Schemas geprüft werden. Dabei wird getestet, ob alle Guidelines und Indikatoren Anwendung finden. Optionale Anforderung ist die Validierung des NBs gegen ein beliebig ausgewähltes Schema.

Veröffentlicht

Solange nicht alle inkludierten Artikel im Zustand „Freigegeben“ sind, kann der NB nicht veröffentlicht werden. Sind alle Artikel im Zustand „Freigegeben“ kann ein Benutzer mit speziellen Rechten den gesamten NB veröffentlichen.

Geschlossen

Ein NB kann in den Zustand „Geschlossen“ gesetzt werden, wenn der NB nicht mehr genutzt werden soll. Dies entspricht einem Entfernen.

5.3.2. Artikel verwalten

Damit Benutzer bei der Erstellung von Artikel innerhalb des NBs unterstützt werden, soll ein Eingabeeditor implementiert werden. Dieser soll die üblichen Funktionen von Web 2.0-Anwendungen anbieten und möglichst komfortabel bedienbar sein. Hieraus ergeben sich Anforderungen, die aktuell bereits von einigen OpenSource-Lösungen adäquat umgesetzt werden. Da eine Neuentwicklung eines solchen Eingabeeditors großen Aufwand bedeuten würde, soll ein OpenSource-Eingabeeditor Produkt verwendet werden.

Abbildung 5.5 verdeutlicht die Anforderungen, die innerhalb des Projektes an einen Eingabeeditor gestellt werden und von diesem erfüllt werden müssen.

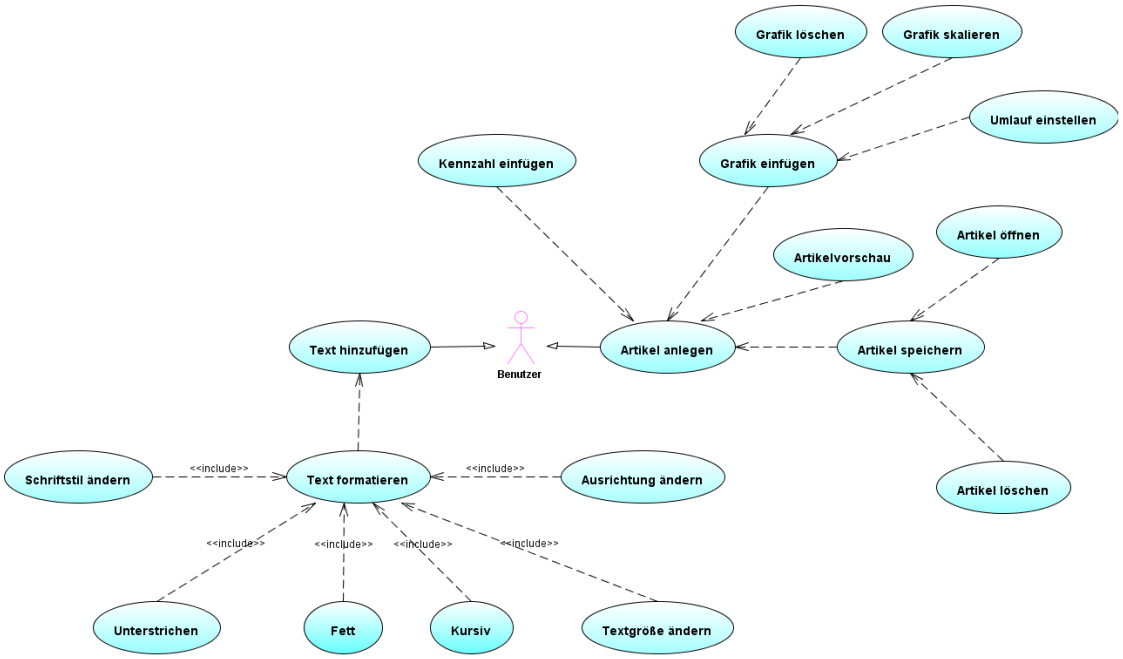

Abbildung 5.5.: Anwendungsfälle im Bereich Eingabeeditor

Anmerkung: Die Begriffe Redakteur und Chefredakteur dienen in diesem Kapitel der Veranschaulichung, orientieren sich aber an dem in Kapitel 5.1.1 vorgestellten Rollenkonzept.

Artikel bilden die inhaltliche Grundlage des Nachhaltigkeitsberichts und werden von Redakteuren über den Eingabeeditor gepflegt. Anforderungen an den Eingabeeditor orientieren sich an den Standard Texteditoren, um die Usability zu erhöhen. So werden für die grafische Repräsentation der Funktionen die typischen Symbole verwendet.

Artikel anlegen

Artikel lassen sich über die Funktion „Neuer Artikel“ im Eingabeeditor anlegen.

Artikel speichern

Artikel können über die Funktion „Speichern“ persistent[3] gespeichert werden. Gespeicherte Artikel stehen auf unbestimmte Zeit bis zu ihrer Schließung bzw. endgültigen Löschung, zur Verfügung. Wird ein Artikel gespeichert, muss dieser einem bestimmten Bereich innerhalb der NB-Struktur zugeordnet werden. Zur Speicherung ist die Angabe einer Artikelüberschrift notwendig.

Artikel öffnen

Artikel können, sofern sie gespeichert worden sind, auch geöffnet werden.

Artikel löschen

Artikel können entfernt werden und gehen damit in den Zustand „Geschlossen“ über.

Artikeltexte formatieren

Die Schriftgröße der Artikeltexte ist frei wählbar. Textteile können Fett, Kursiv und Unterstrichen formatiert werden. Als optionale Anforderung sind weitere Formatierungen möglich.

Textausrichtung ändern

Die Ausrichtung einzelner Textbestandteile innerhalb eines Artikels kann linksbündig, zentriert und rechtsbündig sein und über die jeweilige Funktion festgelegt werden.

Bild einfügen

Über die Funktion „Bild einfügen“ lassen sich Bilder dem Artikel an einer beliebigen Stelle hinzufügen. Unterstützte Formate sind:

- PNG
- JPG
- GIF

Grafik skalieren

Eine Grafik, die hinzugefügt wurde, kann innerhalb des Artikels frei skaliert werden.

Grafik-Umlauf

Der Umlauf einer Grafik legt fest, wie ein Text, der sich auf einer Höhe mit der Grafik befindet, positioniert werden soll.

- Umlauf aus: Der Text wird so ausgerichtet, dass auf der Höhe der Grafik kein Text positioniert wird.
- Seitenumlauf: Der Text wird um die Grafik herum positioniert.

[3] Nicht flüchtige Speicherung beispielsweise in einer Datenbank

Multimedia einfügen

Auch multimediale Inhalte können einem Artikel hinzugefügt werden, über die Funktion „Media hinzufügen“ des Text Editors. Unterstützte Formate sind:

- MP3
- OGG
- WMV
- SWF
- MOV
- AVI
- RM
- MPG
- GIF
- JPG
- PNG

Es folgt eine kurze Erläuterung der Formate:

MP3
Zugehörigkeit: MPEG Audio Stream, Layer III
Klassifizierung: Audio
Mime type: audio/mpeg, audio/x-mpeg, audio/mp3, audio/x-mp3, audio/mpeg3, audio/x-mpeg3, audio/mpg, audio/x-mpg, audio/x-mpegaudio
Quelle und weitere Informationen: http://filext.com/file-extension/MP3

OGG
Zugehörigkeit: Ogg Vorbis
Klassifizierung: Audio
Mime type: Current: audio/ogg; application/ogg, Older: audio/x-ogg; application/x-ogg
Quelle und weitere Informationen: http://filext.com/file-extension/OGG

WMV
Zugehörigkeit: Windows Media File
Klassifizierung: Video, Audio
Mime type: video/x-ms-wmv
Quelle und weitere Informationen: http://filext.com/file-extension/WMV

SWF
Zugehörigkeit: Flash
Mime type: application/x-shockwave-flash, application/x-shockwave-flash2-preview, application/futuresplash, image/vnd.rn-realflash
Quelle und weitere Informationen: http://filext.com/file-extension/SWF

MOV
Zugehörigkeit: QuickTime

Klassifizierung: Video
Mime type: video/quicktime, video/x-quicktime, image/mov, audio/aiff, audio/x-midi, audio/x-wav, video/avi
Quelle und weitere Informationen: http://filext.com/file-extension/MOV

AVI
Zugehörigkeit: Audio Video Interleave File
Klassifizierung: Video
Mime type: video/avi, video/msvideo, video/x-msvideo, image/avi, video/xmpg2, application/x-troff-msvideo, audio/aiff, audio/avi
Quelle und weitere Informationen: http://filext.com/file-extension/AVI

RM
Zugehörigkeit: RealMedia
Klassifizierung: Video, Audio
Mime type: application/vnd.rn-realmedia, audio/vnd.rn-realaudio, audio/x-pn-realaudio, audio/x-realaudio, audio/x-pm-realaudio-plugin
Quelle und weitere Informationen: http://filext.com/file-extension/RM

MPG
Zugehörigkeit: MPEG 1 System Stream
Klassifizierung: Video, Audio
Mime type: video/mpeg, video/mpg, video/x-mpg, video/mpeg2, application/x-pn-mpg, video/x-mpeg, video/x-mpeg2a, audio/mpeg, audio/x-mpeg, image/mpg
Quelle und weitere Informationen: http://filext.com/file-extension/MPG

GIF
Zugehörigkeit: Graphic Interchange Format
Klassifizierung: Grafik, Bild
Mime type: image/gif, image/x-xbitmap, image/gi_
Quelle und weitere Informationen: http://filext.com/file-extension/GIF

JPG
Zugehörigkeit: JPEG/JIFF Image
Klassifizierung: Grafik, Bild
Mime type: image/jpeg, image/jpg, image/jp_, application/jpg, application/x-jpg, image/pjpeg, image/pipeg, image/vnd.swiftview-jpeg, image/x-xbitmap
Quelle und weitere Informationen: http://filext.com/file-extension/JPG

PNG
Zugehörigkeit: Portable (Public) Network Graphic
Klassifizierung: Grafik, Bild
Mime type: image/png, application/png, application/x-png
Quelle und weitere Informationen: http://filext.com/file-extension/PNG

Kennzahlen

Kennzahlen (Indikatoren) werden in dem Eingabeeditor in einer sogenannten Kennzahlenliste (Indikatorenkatalog, definiert im NB-Schema) zusammengefasst. Über diese Kennzahlenliste können einzelne Kennzahlen im Artikel positioniert werden.

Artikelvorschau

Eine Artikelvorschau ist sowohl über WYSIWYG[4] als auch über WYSIWYM[5] möglich. Eine optionale Anforderung ist es, einen Artikel, der sich in der Bearbeitung befindet und im Eingabeeditor geöffnet ist, über die Funktion „Artikelvorschau in PDF“ in ein Endergebnis im Format PDF anzeigen zu lassen.

Ein einzelner Artikel kann sich in den folgenden Zuständen befinden.

In Bearbeitung

Artikel in diesem Zustand können von Redakteuren bearbeitet und von Chefredakteuren geschlossen oder freigegeben werden. „In Bearbeitung“ ist der Initialzustand[6] eines jeden Artikels.

Freigabe

Befindet ein Chefredakteur einen Artikel für fertig, kann er diesen freigeben. Der Artikelzustand wechselt dann von „In Überprüfung“ zu „Freigegeben“. Nur ein Benutzer mit speziellen Rechten hat nach der Freigabe noch die Möglichkeit, den Artikel wieder zurück in die Bearbeitung zu reichen.

Geschlossen

Durch den Zustand „Geschlossen“ werden Artikeln, die sich noch „in Bearbeitung“ befinden, nicht weiter berücksichtigt. Artikel, die bereits in einem publizierten NB stehen, können nur durch einen Benutzer mit speziellen Rechten geschlossen werden und tauchen dadurch nicht mehr in dem NB auf.

[4] What you see is what you get; Dokument wird während der Bearbeitung genauso angezeigt wie es bei der Ausgabe aussieht

[5] What you see is what you mean; stellt den Text am Bildschirm so dar, dass hervorgeht, welchen Zweck eine Formatierung erfüllen soll

[6] Zustand, der einem Artikel im Moment der Erstellung zugeordnet wird

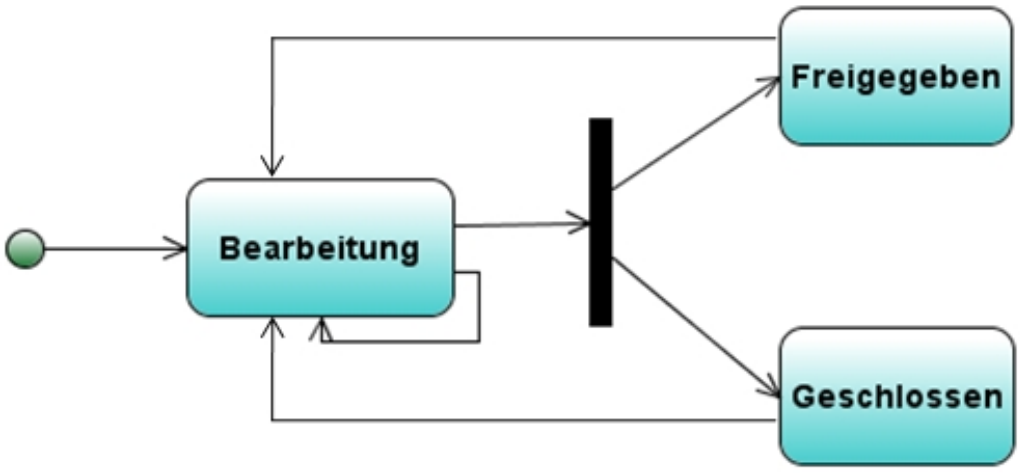

Abbildung 5.6.: Eingabeeditor: Artikelzustände

5.4. Ausgabe von Nachhaltigkeitsberichten

Nachdem die Struktur und Inhalte des NB im System definiert wurden, hat der Benutzer die Möglichkeit den erstellten Bericht zu betrachten. Die Ausgabe des NBs erfolgt dabei auf drei unterschiedliche Arten:

1. Website
2. PDF
3. XML

Grundsätzlich unterscheiden sich die verschiedenen Ausgabeformate in der Art ihrer nachfolgenden Nutzung. Während eine Anzeige in Form einer Website oder eines PDF-Dokuments der anschaulichen Aufbereitung für den Benutzer dient, ist das XML-Format vorteilhaft für die maschinelle Verarbeitung des Nachhaltigkeitsberichtes um z.B. mehrere Nachhaltigkeitsberichte miteinander vergleichen zu können. Abbildung 5.7 gibt hier eine Übersicht über die Möglichkeiten eines Benutzers sich einen NB anzuzeigen.

Die einzelnen Ausgabeformate werden im Folgenden näher beschrieben.

5.4.1. HTML

Um die Nachhaltigkeitsberichte strukturiert auf einer Website - sprich in HTML - ausgeben zu können, wird diese in mehrere Bereiche aufgeteilt.

Navigationsleiste

Die Navigationsleiste stellt die Struktur der NBs dar. Über geschachtelte Haupt- und Unterkategorien sind sämtliche Kapitel oder Artikel des NBs per Link zugänglich. Zur besseren Übersicht werden Unterkapitel erst sichtbar, sobald die Oberkapitel vom Benutzer gewählt wurden.

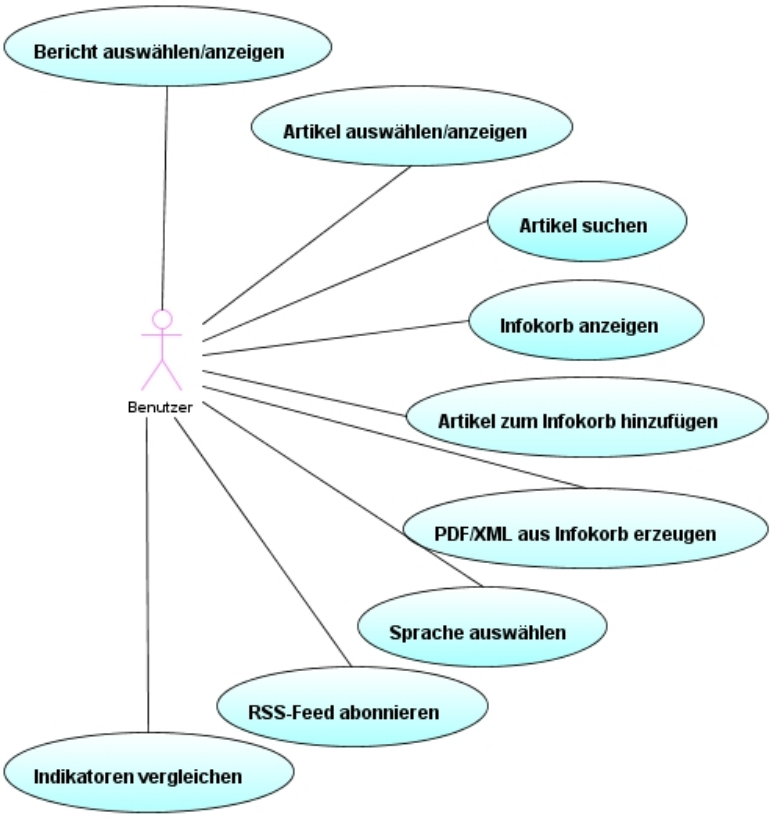

Abbildung 5.7.: Anwendungsfälle im Bereich Ausgabe

Auf oberster Ebene werden sämtliche Nachhaltigkeitsberichte der vergangenen Jahre absteigend sortiert angezeigt.

Inhaltsseite

Die Inhaltsseite präsentiert den eigentlichen Inhalt eines einzelnen Artikels eines NBs. Dies können neben reiner textueller Darstellung ebenso multimediale Inhalte sein.

Eine Breadcrumb-Navigation /footnoteNavigationselemente, die den Pfad zum aktuellen Element zeigen oberhalb der Inhaltsseite zeigt die Position des jeweiligen Kapitels innerhalb der Gesamtstruktur und ermöglicht eine einfachere Navigation. Diese Funktionalität ist optional.

Kopfbereich

Der Kopfbereich befindet sich oberhalb der Inhaltsseite und trägt ein frei wählbares Logo. In dem Falle der PG STORM wird hier das Logo der Universität Oldenburg verwendet. Ebenso kann ein Seitentitel frei gewählt werden. Darüberhinaus gibt es Links zur Startseite, Hilfe und zur Sprachauswahl. Im Kopfbereichgibt es weiterhin ein Suchfeld, das allgemeine Suche, NBs, die Artikel eines NBs oder Kommentare zu einem Artikel anhand eines Suchbegriffes ermöglicht.

Fußbereich

Der Fußbereich befindet sich unterhalb der Inhaltsseite. Dieser bietet Platz für Weiterleitungen auf Seiten mit zusätzlichen Funktionen oder zu weiteren Informationen. Unter anderem kann der Anwender sich hier für den Newsletter von Storm registrieren, die Sitemap und das Impressum einsehen, sowie die Kontaktseite zu betrachten. Gerade die Impressumseite und die Kontaktseite sind für eine Anwendung zur Nachhaltigkeitsberichterstattung von besonderer Wichtigkeit, da jegliche Webseite gewisse gesetzliche Aspekte entsprechend der Art der Instituts- oder Unternehmensart erfüllen muss. Hierzu gehören zum Beispiel die Kontaktdaten der Insitution oder des Unternehmens oder Informationen über Aufsichtsbehörden. Zudem ist die Kontaktseite notwendig, damit Anwender Kontakt zur Institution oder zum Unternehmen, welche den NB vorhält, aufnehmen

5.4.2. PDF

Der Benutzer kann jeden einzelnen Artikel nicht nur online ansehen, sondern sich diesen auch als PDF (Portable Document Format) [7] anzeigen und ausdrucken lassen. Desweiteren erlaubt es die Anwendung den Inhalt eines eigen über den Infokorb zusammengestellten Berichts als PDF herunterladen und ausdrucken zu können. Dem jeweils generierten PDF-Bericht wird automatisch ein zugehöriges Inhaltsverzeichnis vorangestellt. Dieses listet dann die einzelnen ausgewählten Artikel mit zugehöriger Seitenzahl auf. Die Funktion des generierten Inhaltsverzeichnisses ist dabei jedoch optional.

[7] Das Portable Document Format (PDF) gibt elektronische Dokumente unabhängig von dem verwendeten Betriebssystem und der Hardware original wieder.

5.4.3. XML

Neben der Möglichkeit, Informationen als PDF anzeigen zu lassen, ist auch die Anzeige als XML (Extensible Markup Language)[8] möglich. Die abfrage von informationen als XML dient der maschinellen Weiterverarbeitung. Optional soll ein Webservice zur Verfügung gestellt werden, um bestimmte Artikel (oder ganze NBs) in XML-Darstellung extern maschinell abzurufen. Einzelne Artikel könnten dann durch verschiedene Suchkriterien extrahiert werden.

5.5. Usability-Funktionen

Um dem Benutzer ein möglichst komfortables System zu liefern, soll der Usability (deutsch: Bedienbarkeit, Benutzbarkeit) eine erhöhte Aufmerksamkeit während der Erstellung des Endproduktes zugewiesen werden. Die Anforderungen wurden hierbei unter Berücksichtigung der in den „Web Content Accessibility Guidelines 1.0“ formulierten Richtlinien des World Wide Web Konsortiums erarbeitet.

5.5.1. Navigation

Das Navigationssystem fasst die Bereiche des Webangebots in logisch voneinander abgrenzbare Teile zusammen. Bereiche die bereits besucht wurden, werden als solche durch eine farbliche Anpassung kenntlich gemacht. Darüber hinaus ist für den Benutzer zu jeder Zeit erkennbar, wo er sich innerhalb des Webangebots befindet. Um die Startseite von jedem Bereich des Webangebots erreichen zu können, ist das Logo mit der Startseite des Webangebots verlinkt.

5.5.2. Internationalisierung

Die Ausgabe der Nachhaltigkeitsberichte soll mehrsprachig gestaltet werden können. Das heißt einzelne Artikel können mehrsprachig eingegeben und angezeigt werden um auch nicht deutschsprachigen Benutzern die Bedienung des Systems zu ermöglichen. Dabei liegt es in der Verantwortung der Redakteure die Mehrsprachigkeit in Bezug auf die Inhalte zu gewährleisten. Die zu entwickelnde Software wird keine Übersetzungsfunktionen bereitstellen. Das Softwareprodukt wird in deutscher und englischer Sprache ausgeliefert.

5.5.3. Validierung von Benutzereingaben

Benutzerinteraktionen finden in verschiedenen Bereichen der Webanwendung statt. Um Benutzer bei der Interaktion mit der Webanwendung zu unterstützen, werden alle Benutzereingaben - sofern sie überprüft werden können - validiert[9]. Benutzereingaben werden zunächst clientseitig, dann serverseitig validiert. Auf eine clientseitige Validierung kann verzichtet werden, sofern dies technisch nicht möglich ist, bspw. wenn eine clientseitige Validierung eine serverseitige Validierung voraussetzt. Auf eine Validierung bestimmter

[8] Die Extensible Markup Language (XML) ist eine Auszeichnungssprache zur Erstellung von Dokumenten, die maschinell verarbeitet werdenkönnen.

[9] Validierung von Benutzereingaben entspricht der Überprüfung auf Korrektheit und Vollständigkeit.

Benutzereingaben kann vollständig verzichtet werden, wenn die zu validierenden Benutzereingaben maschinell nicht geprüft werden können oder eine Validierung zu kostenintensiv [10] ist.

Abbildung 5.8 fasst die Anforderungen an die Usability der Webanwendung zusammen.

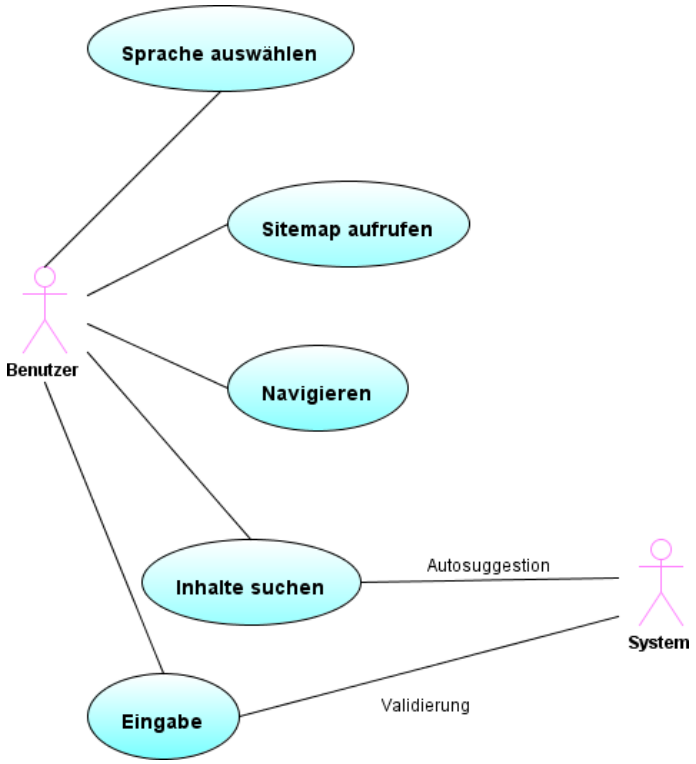

Abbildung 5.8.: Anwendungsfälle im Bereich Usability

[10] Hoher Rechenaufwand, hoher maschineller/personeller Aufwand.

5.6. Personalisierung

Die Ausgabe der Nachhaltigkeitsberichte soll personalisiert werden können. Allgemein wird unter Personalisierung die Integration der Nutzerdaten verstanden, die sowohl zur persönlichen Ansprache des Anwenders, als auch zur zielgerichteten Inhaltsempfehlungen des Systems (siehe hierzu Abschnitt 5.6.1) an den Benutzer, bzw. die Möglichkeit zur Hinterlegung von speziellen Benutzerdaten (Historie) dienen.

Durch die Personalisierung kann der Benutzer namentlich begrüßt werden, da Daten über den Benutzer hinterlegt und zur Erstellung eines Benutzerprofils eingesetzt werden. Ebenfalls ist ein Infokorb mit der Möglichkeit der individuellen Auswahl von Informationen - Artikeln - und Speicherung dieser zur späteren Nutzung, realisierbar.

Um auf Veränderungen oder Neuerscheinungen im Interessenbereich des Benutzers hinzuweisen zu können, eignen sich RSS-Feeds (siehe Abschnitt 5.7.7).

Der Bereich Personalisierung behandelt dabei den Umgang mit den persönlichen Daten der Benutzer und umfasst zugleich die persönlichen Benutzerdaten, den Infokorb und selbst erstellte Nachhaltigkeitsberichte. Die persönlichen Daten sind standardmäßig nur für den registrierten Benutzer einsehbar.

5.6.1. Vorschlagswesen

Innerhalb des Inhaltsbereiches (auf der Startseite der Anwendung) bietet Storm ein Vorschlagswesen (VW). Dieses unterstützt den Anwender in der Benutzung der Anwendung, indem es verschiedene Artikel vorschlägt. Dabei werden die Vorschläge aus einer anwenderspezifischen Auswahl von betrachteten Artikeln erstellt. Grundsätzlich bietet das VW folgende Möglichkeiten:

- Übersicht über beliebteste Artikel
- Zuletzt betrachtete Artikel
- Weitere interessante Artikel
- Artikel, die für die entsprechende Interessengruppe von Interesse sind
- Artikel, die von befreundeten Anwendern betrachtet wurden

Um eine Übersicht über beliebte Artikel zu bieten berechnet Storm aus der Anzahl der aufgerufenen Artikel eine Relevanz und gibt die beliebtesten Artikel aus. Diese Möglichkeit ist dem Anwender jederzeit möglich und steht auch einem nicht angemeldeten Anwender zur Verfügung. Ist der Anwender angemeldet, bietet Storm weitere Möglichkeiten. So wird der Anwender über eine Auswahl zuletzt betrachteter Artikel informiert um schnell wieder auf Artikel zugreifen zu können. Des Weiteren wird der Anwender auch vom VW direkt in der Benutzung von Storm unterstützt, indem das Vorschlagswesen aus dem allgemeinen Anwenderverhalten weitere für den speziellen Anwender oder für die Zielgruppe des Anwenders interessante Artikel auswählt und Artikel, die von befreundeten Anwendern betrachtet wurden anzeigt.

5.6.2. Fragen-System

Das Fragen-System der Anwendung bietet dem Benutzer die Möglichkeit Fragen zu stellen, Fragen zu kommentieren und Antworten zu bestehenden Fragen zu lesen. Das System ist dem Prinzip von „Direktzu“ nachempfunden [Dir09]. Diese Anforderung ist optional.

5.6.3. Template-Auswahl

Um das Erscheinungsbild der Anwendung ändern zu können wird ein Template-System implementiert, welches es ermöglicht zwischen verschiedenen Templates zu wechseln bzw. neue Templates zu aktivieren. Als Template wird in diesem Zusammenhang ein spezielles Design von Storm bezeichnet. Es ermöglicht die Anpassung des Systems an die Gegebenheiten und das Corporate Design der Institution bzw. des Unternehmens.

5.6.4. Auswertung von Benutzerprofilen

In expliziten Benutzerprofilen werden benutzerspezifische Informationen vom Benutzer direkt selber gepflegt und sind relativ statisch. Dazu gehört z.B. die Angabe einer Interessentenkategorie (Stakeholder). Diese umfassen z.B. Professoren, Studenten, Mitarbeiter oder ähnliche. Auch eine Angabe der Organisation des Benutzers (z.B. Uni Oldenburg, bestimmte Firmen, o.ä.) kann zu sinnvollen Analysen beitragen. Zu beachten ist hier, dass Benutzer auch bewusst nicht richtige Informationen angeben können, und damit eine Analyse ggf. nicht mehr aussagekräftig sein kann.

Implizite Benutzerprofile entstehen durch das Sammeln von Informationen über den Verlauf der Benutzerbesuche. Dazu zählen die besuchten Seiten, die heruntergeladenen Inhalte und bewertete oder getaggte Inhalte, die einem bestimmten Benutzer zuzuordnen sind. Die Analyse der in der Anwendung genutzten Benutzerprofile stellt eine optionale Anforderung dar. Die Auswertung von Benutzerprofilen kann dazu dienen Benutzer zu klassifizieren, bzw. andere Benutzer mit ähnlichen Eigenschaften zu finden.

5.7. Weitere Anforderungen

Ziel des Projektes ist es ein System zu schaffen, dass Dialogorientierung zwischen den Nutzern des Systems einerseits und andererseits der Anwendung selbst unterstützt. Um dies zu erreichen sind weitere Anforderungen an das System zu stellen, um eben diese Funktionalität zu erlauben.

5.7.1. Autovervollständigung von Suchbegriffen

Ein Anforderungen zur Unterstützung des Anwenders innerhalb des Systems bietet die Autovervollständigung. Autovervollständigung verfolgt dabei das Prinzip dem Anwender schon im Verlauf der aktuellen Eingabe verschiedene Wörter oder Phrasen vorzuschlagen und so im Sinne der Usability (siehe Abschnitt 5.5 in der Bedienung zu unterstützen. Diese Option ist auch deshalb interessant, da sie die Möglichkeit schafft, die Interaktionen zwischen System und Benutzer zu beschleunigen.

5.7.2. Bewertung von Inhalten

Einzelne Inhalte können bewertet werden, das heißt Benutzer können Bewertungen für die Relevanz oder Qualität der Inhalte vergeben. Auf Basis dieser Bewertungen lassen sich dann Rankings der Inhalte generieren, die sowohl global als auch auf bestimmte Bereiche ausgerichtet sein können.

5.7.3. Hilfefunktionen

Um dem Benutzer die Funktionen, die der Benutzer verwenden kann, zu erkären, so dass der die Funktionen korrekt einsetzen kann, gibt es die Hilfefunktion. Ein Benutzer, der beispielsweise keinen digitalen Infokorb kennt, könnte sich in der Hilfe die Beschreibung des Infokorbs durchlesen und weiss anschließend, was der Infokorb ist und kann diesen benutzen.

5.7.4. Kommentarfunktion

Ein Hauptmerkmal des Systems stellt die Dialogmöglichkeit mit den Lesern bzw. Benutzern der Webanwendung dar. Statt - wie in der Praxis bisher üblich - den NB nur zu präsentieren, sollen Benutzer die Möglichkeit erhalten, in einen Dialog mit den Verfassern des NBs und anderen Benutzern zu treten. Zu diesem Zweck soll das Endprodukt eine Kommentarfunktion anbieten, die es Benutzern erlaubt, zu einzelnen Artikeln Kommentare zu verfassen, bearbeiten und speichern. Die Kommentarfunktion wird auch zu den Kommentaren selber angewendet in dem eine hierarchische Struktur von belibig vielen Kommentare aufgebaut werden kann.

5.7.5. RSS-Import

Um externe RSS-Feeds (z.B. News) in die Anwendung zu integrieren, besteht die Möglichkeit einen oder mehrere RSS-Feeds zu importieren. Diese Anforderung ist optional.

5.7.6. Newsletter

Um dem Benutzer die Möglichkeit zu geben, sich aus der Anwendung heraus an einem Newsletter anzumelden, können externe Newsletter-Systeme integriert werden. Diese Anforderung ist optional

5.7.7. RSS-Feed

Um interessierte Benutzer über Neuerungen innerhalb der NBs zu informieren, gibt es die sogenannten RSS-Feeds. Diese Funktion ist optional.

RSS ist ein plattformunabhängiges und auf XML basierendes Format, das entwickelt wurde, um Nachrichten und andere Webinhalte auszutauschen. Die Abkürzung RSS steht dabei für „Really Simple Syndication“.

Mit RSS ist es möglich dem Benutzer - der die RSS-Feeds abboniert hat - schnell und effektiv auf Änderungen hinzuweisen oder über aktuelle Neuigkeiten zu informieren. Diese Funktionalität soll optional auch in der im Projekt zu entwickelnden Software realisiert werdenn. Es können spezielle RSS-Feeds für die unterschiedlichen Stakeholdergrupen angeboten werden.

5.7.8. TagCloud

Storm bietet den Anwender die Möglichkeit auf allen Artikelseiten Schlagwörter (sogenannte Tags) einzugeben. Diese Schlagwörter werden intern gespeichert und dienen einer Schlagswortwolke (TagCloud) als Basis. „Tag Clouds“ ist eine Technologie die mittels einer Liste von Tags (Stichwörter) Daten visualisiert. Dabei werden Informationen entsprechend ihrer Wichtigkeit angeordnet. Durch Cascading-Style-Sheets (CSS) können Tag Clouds zudem beliebig formatiert werden, um die Häufigkeit und Wichtigkeit der Daten prägnant darzustellen.

Wie in Abbildung 5.9 ersichtlich können Schlagwörter, die von höherer Wichtigkeit sind, grösser und fetter dargestellt werden, während Schlagwörter mit niedrigerer Bedeutung kleiner und in schwächerer Farbe dargestellt werden.

5.7.9. Trackback

Bei Trackback handelt es sich um eine Form der automatisierten Kommunikation zwischen Webdokumenten. Dabei werden Informationen mittels gegenseitigen Informationsaustauschs zwischen Webseiten weiter gereicht.

Der Hintergrund und Sinn von Trackbacks besteht darin, dass sowohl der Autor eines Artikels als auch dessen Leser darüber informiert werden, dass es an einer anderen Stelle im Internet weitere Informationen gibt, die von Interesse sein könnten. Grundsätzliches Ziel ist es immer die Leser beider Webseiten über die jeweils andere Informationsquelle zu informieren. Die Trackback-Funktionalität ist eine optionale Anforderung. Ein Trackback erklärt am Beispiel einer Pressemeldung ist in Abbildung 5.10 zu sehen.

Abbildung 5.9.: Tag Cloud Beispiel Web 2.0 [Quelle: [Ang05]]

Abbildung 5.10.: Trackback am Beispiel einer Pressemeldung ([Ver09])

5.7.10. Startseite

Beim Aufruf der Software STORM und nach erfolgter Anmeldung wird der Benutzer auf die Startseite weitergeleitet. Auf dieser Startseite sollen neben einem Begrüßungstext auch einige Vorschläge und beispielsweise die neuesten Kommentare angezeigt werden.

6. Nicht funktionale Anforderungen

Neben den rein funktionalen Anforderungen an das Softwareprodukt sollen auch Anforderungen erfüllt werden, die nicht direkt mit der Arbeitsweise des späteren Systems in Verbindung gebracht werden können. Diese nicht funktionalen Anforderungen sind Qualitätseigenschaften, die eine Software erfüllen muss. Die wichtigsten nicht funktionalen Anforderungen, die das Softwareprodukt erfüllen soll, sind im Folgenden näher beschrieben.

Modularität/Flexibilität

Da die Software später grundsätzlich auch für den Einsatz in beliebigen Organisationen verwendbar sein soll, muss sie modular gestaltet werden. Modularität bedeutet hierbei dass es ermöglicht werden muss, dass bestimmte Funktionalitäten des Systems getrennt voneinander (in Modulen) zu betreiben. Um den Anforderungen nach einem hohen Funktionsumfang oder einer, speziell im Hinblick auf Sicherheitsaspekte wichtigen, Kontrollierbarkeit der Anwendung entgegen zu kommen soll Storm dahermit einer unkomplizierte Erweiter- oder Abschaltbarkeit von Modulen ausgestattet werden.

Sprache

Sämtliche Bezeichner und Kommentare im entwickelten Quellcode sollen in englischer Sprache gehalten werden. Dies soll, unterstützt durch die globale Verbreitung des englischen Sprachraums die Möglichkeit bieten, eine weiterentwicklung auch durch Personen anderer Nationalitäten zu ermöglichen. Desweiteren soll die Bedienung von Storm in diversen Sprachen möglich sein.

Korrektheit

Das Endprodukt des Projektes soll neben der korrekten Umsetzung der funktional und nicht funktionalen Anforderungen stets ein wieder reproduzierbares, vollständiges und korrektes Ergebnis liefern. Dies ist insbesondere wichtig, um eine spätere Fehlerbehandlung der Software zu ermöglichen.

Sicherheit

Zur Sicherheit des Produktes gehören vor allem die zwei Aspekte Datenschutz und Verhinderung von Code Injection.

Datenschutz

Der Datenschutz innerhalb der Software muss nach den rechtlichen Bestimmungen gewährleistet sein. So sollen z.B. keine benutzerspezifischen Daten herausgegeben werden können und das System soll nicht von unbefugten Personen manipulierbar sein. Eine explizite Datenschutzerklärung muss aus diesen Gründen vorhanden sein. Eine Unterstützung des

P3P-Standards[1] erscheint dabei sinnvoll, wird aber als optional betrachtet. Um die Anforderung an die Datensicherzeit zu gewährleisten, müssen personenbezogene Daten nach gängigem Standard (SSL) verschlüsselt übertragen werden können.

Verhinderung von Code Injection

Code Injection ist eine Methode, fremden Programmcode (häufig SQL- oder Java Script Anweisungen, aber auch andere sind möglich) in das Produkt einzuschleusen, der in der Regel eine schädliche Funktion hat. Üblicherweise wird Code Injection an Stellen eingesetzt, an denen Texteingaben möglich sind (z.B. in einem Kommentarfeld). Die zu erstellende Software wird in der Endversion keine Möglichkeit zur Code Injection mehr bieten.

Browserunabhängigkeit

Das Softwareprodukt soll alle gängigen Browser (Mozilla Firefox, Opera, Internet Explorern, Safari, Crome) gleichermaßen unterstützen.

Gestaltung

Die Anwendung sollte nach den üblichen Richtlinien zur Benutzbarkeit so gestaltet sein, dass sie intuitiv bedienbar ist und alle Elemente übersichtlich angeordnet und gut zu erkennen sind. Beispielsweise soll es auch bei deaktiviertem JavaScript möglich sein, alle Inhalte, wenn auch in anderer Darstellung, abzurufen. Nicht sofort verständliche Funktionen sollen zusätzlich mit erklärenden Hinweisen versehen sein. Zudem soll auf spezielle Aspekte geachtet werden. Hierzu gehören unter anderem eine Orientierung an der Endbenutzer-Zielgruppe, aber auch eine zielgerichtete Ausrichtung auf den späteren Workflow und eine Konformität mit modernen Webstandards. Bereits im Standard-Template soll auf die Leseweise innerhalb einer Webanwendung geachtet werden. Dies bedeutet, dass der natürliche Lesefluss, der in Z-Form durchgeführt wird, bereits durch das Template unterstützt wird.

Zuverlässigkeit

Die Anwendung soll dem Benutzer ständig zur Verfügung stehen. Ausgenommen sind Wartungsarbeiten und Ausfälle wegen höherer Gewalt. Wartezeiten (z.B. bei Ladevorgängen) sind so gering wie möglich zu halten. Sollte es zu einem Fehler innerhalb der Software kommen, muss der Benutzer informiert werden, dass ihm die Anwendung zeitweise nicht zur Verfügung steht.

[1] Platform for Privacy Preferences, siehe z.B. http://de.wikipedia.org/wiki/Platform_for_Privacy_Preferences

Teil III.

Entwurf

7. Vorbemerkungen zum Entwurf

Die in diesem Teil III vorliegende Entwurfsspezifikation beschreibt im Detail, wie die zu entwickelnde Software mit den in der Anforderungsdefinition ermittelten Anforderungen zur Erstellung und Darstellung von Nachhaltigkeitsberichten implementiert werden soll. Es sind grundlegende Entwurfsentscheidungen zur Architektur, Komponenten, Schnittstellen und deren Charakteristika begründet dargelegt.
Die Implementierung der, in der Anforderungsdefinition dargestellten Anwendungsfälle wird in den jeweiligen Kapiteln detailliert beschrieben.
Innerhalb dieser Einleitung werden zunächst grundlegende Konzepte und Sachverhalte - die im Rahmen des Projekts eine Rolle spielen - erläutert, wonach es in den folgenden Kapiteln konkret um die Systemarchitektur (Kapitel 8) und um die Systemkomponenten (Kapitel 9) geht. Grundsätzlich wird im Rahmen der Softwareentwicklung auf das Architekturmuster Model View Controller (MVC) gesetzt. Dieses Architekturmuster strukturiert die Entwicklung in die drei Bereiche: Model (Datenmodell), View (Präsentation) und Controller (Programmsteuerung). Das Model enthält die darzustellenden Daten. Die View ist für die Darstellung der Daten und die Entgegennahme von Benutzertransaktionen zuständig. Der Controller wertet die Benutzeranfragen aus und agiert entsprechend. Durch dieses Architekturmuster wird ein flexibler Programmentwurf gewährleistet, der eine spätere Änderung und Erweiterung ermöglicht sowie eine mögliche Wiederverwendbarkeit sicherstellt.

Zur Modellierung - im Rahmen der erstellten Diagramme zur Verdeutlichung einzelner Verhaltensweisen oder Sachverhalte - dienen die Grundsätze ordnungsgemäßer Modellierung (GoM) als Basis. In den GoM sind Richtlinien formuliert, die die Sicherstellung der Qualität von Informationsmodellen - in diesem Fall (UML-)Diagramme - adressieren. Dabei gehen die GoM über die reine syntaktische Korrektheit hinaus. Innerhalb der GoM werden sechs Kriterien als die Wesentlichen gekennzeichnet, diese sind (siehe [Bec]):

- Grundsatz der Richtigkeit
- Grundsatz der Relevanz
- Grundsatz der Wirtschaftlichkeit
- Grundsatz der Klarheit
- Grundsatz der Verständlichkeit
- Grundsatz des systematischen Aufbaus

Weiterhin stellen die Begriffe Frontend und Backend und das, hinter diesen Begriffen stehende Konzept, einen wesentlichen Aspekt dar. Frontend und Backend sind Begriffe, die in der IT verwendet werden, um eine Schichteneinteilung zu realisieren. Das Frontend ist typischerweise näher am Benutzer, das Backend näher am System. Das Frontend beinhaltet alle Views - View angelehnt an das MVC-Konzept. Im Bezug auf die zu entwickelnde Software, stellt das Frontend beispielsweise den Eingabeeditor, den Schemaeditor oder den Infokorb (Details folgen in Kapitel 9) dar. Das Backend betrifft den Controller und das Model aus dem MVC Architekturmuster.

8. Systemarchitektur

In diesem Kapitel wird die Architektur des Softwaresystems beschrieben und alle Komponenten dieser Architektur erläutert.

8.1. Server

Als Anwendungsserver wird der JBoss für das Entwicklungs- und Testsystem genutzt. JBoss ist eine Implementierung eines Anwenungsservers nach dem Java-EE-Standard und ein Open Source Produkt.

Folgende Gründe sprechen für den Einsatz von JBoss:

- Java EE Support, relavant für die Verwendung von Grails
- Open-Source
- zertifizierter Applikationsserver
- Stabilität gegenüber anderen Plattformen
- höhere Entwicklerproduktivität
- Standardkonformität
- gute Integration mit weiteren Open-Source-Projekten

8.2. Datenbankmanagementsystem

Als Datenbank wird während der Entwicklungszeit MySQL eingesetzt. Die Gründe für diese Wahl sind folgende:

- Plattformunabhängigkeit
- Open-Source
- kostenlos
- hohe Performanz
- SSL-Verschlüsselung
- einfache Administration
- Stabilität
- bereits gute Kenntnisse innerhalb des Entwicklerteams

8.3. Grails

Eine Anforderung an die Projektgruppe war das Framework[1] Grails zu benutzen. Grails ist ein Open Source Framework, welches auf etablierten Java-Frameworks basiert und diese mit den Eigenschaften der dynamischen Programmiersprache Groovy - ebenfalls Java-basierend, verbindet.

Die wichtigsten Komponenten von Grails sollen im Folgenden näher erläutert werden.

Hibernate

Hibernate ist ein Open Source Persistenz-Framework für Java und bietet durch das Object Relation Mapping (ORM), die Möglichkeit, Objekte mit ihren Attributen und Operationen auf relationalen Datenbank-Tabellen abzubilden, auf diese einfach und zuverlässig zuzugreifen und aus bereits bestehenden Datenbank-Tabellen Objekte zu erzeugen. Grails nutzt Hibernate für das OR-Mapping (Object-Relational-Mapping)

Spring

Spring ist ein Open Source Applikations-Framework für Java-Plattformen und unterstützt das MVC-Konzept. Grails baut auf dem Spring-MVC auf, indem es das Spring MVC als das darunterliegende Web-Applikationsframework nutzt und dieses erweitert.

Model Das Model wird in Grails durch Domain-Klassen repräsentiert. Domain-Klassen ähneln Java-beans. Ein wesentliches Merkmal ist die ausschließliche dekleration von Attributen und Bedingungen, die zu diesen Attributen deklariert werden können. Jede Domain-Klasse wird durch grails und Hibernate auf ein relationales Datenmodell abgebildet und mit diesem in Beziehung gesetzt.

View Die View-Komponente in Grails wird durch Groovy-Server-Pages (GSP) repräsentiert. GSP bestehen sowohl aus HTML- als auch aus Java- bzw. Groovy-Code. Wird eine GSP angefordert, wird der Groovy-Code ausgeführt.

Controller Ein Controller in Grails ist für die Bearbeitung von Anfragen und die Erstellung oder Aufbereitung von Antworten zuständig. Controller in Grails sind im Wesentlichen Klassen, bestehend aus Java- und/oder Groovy-Code. Jeder Controller verfügt über eine Menge von Actions (deutsch „Aktionen“). Diese bilden die eigentliche Programmlogik. Jeder Controller verfügt dank Vererbung über eine Reihe von Eigenschaften, die dynamisch zur Laufzeit mit Werten gefüllt werden und Entwicklern zur Verfügung stehen.

Bei der Entwicklung von Grails spielten einige Prinzipien eine besondere Rolle, die die Entwicklung von Webanwendungen maßgeblich beeinflussen und eine effizientere Entwicklung ermöglichen. Eine Auswahl rudimentärer Prinzipien wird nachfolgend erläutert:

Scaffolding Scaffolding bietet dem Entwickler die Möglichkeit, zu einer bestehenden Domain-Klasse, ein grundlegendes Gerüst einer Anwendung automatisch zu generieren.

Convention over Configuration (CoC) Das CoC-Prinzip bietet vordefinierte Standards an, durch die den Entwicklern der Großteil an Konfigurationsaufwand abgenommen wird und diese sich somit auf die eigentliche Programmierung konzentrieren können.

[1] Ein Framework (englisch für „Rahmenstruktur“): Bietet ein Programmiergerüst und stellt Softwarentwicklern wesentliche Grundfunktionalitäten zur verfügung.

Don't Repeat Yourself (DRY) „jedes Stück Wissen eine einzige, eindeutige und massgebliche Repräsentation in einem System hat“ [AH03]. Nach dem DRY-Prinzip sollten folglich Redundanzen vermieden werden. Als „ein Stück Wissen“ können Daten, Funktionalitäten oder auch Logik gesehen werden [MB09].

8.4. Client

STORM ist eine Webanwendung und wird über einen Browser[2], der auf einem Client installiert ist bedient, wodurch die Nutzung der Anwendung plattformunabhängig ist. Theoretisch kann die Anwendung über jedes System, dass über einen Browser verfügt, bedient werden. Da die Anzeige von HTML in Browsern jedoch nur schwach standardisiert ist, kann eine Browserunabhängigkeit nicht garantiert werden. Die Darstellung kann von Browser zu Browser variieren.

[2] Browser (oder engl. to browse 'schmökern', 'umsehen', auch 'abgrasen') sind spezielle Computerprogramme zur Darstellung von Webseiten im World Wide Web.[Unkb]

8.5. Architektur-Überblick

Um die Teilkomponenten der Architektur besser in den Gesamtzusammenhang einordnen zu können, ist in Abbildung 8.1 die Systemarchitektur nochmal übersichtlich dargestellt.

Der Client stellt über HTTP Anfragen an den Anwendungsserver. Antworten des Anwendungsservers können wahlweise PDF, XML oder HTML sein. PDF bspw., wenn der Client einen Nachhaltigkeitsbericht als PDF angefordert hat. Anfragen, die der Anwendungsserver entgegennimmt, werden von Grails an den zu verarbeitenden Controller innerhalb der Applikation (Application) gereicht. Die für die Bearbeitung einer Anfrage benötigten Daten, werden über Hibernate aus der MySQL-Datenbank angefragt und auf Objekte (Instanzen von Domain-Klassen) abgebildet. Externe Prozesse (external Processing) können Daten als XML (siehe Kapitel 9.3.7) (XBRL optional) anfragen. Aus externen Systemen (external Systems) wie z.B. SAP (siehe Kapitel 9.2.2 Externe Systeme) können Daten über eine einheitliche Datenschnittstelle (Unified Data Layer) in die Applikation (Application) geladen werden.

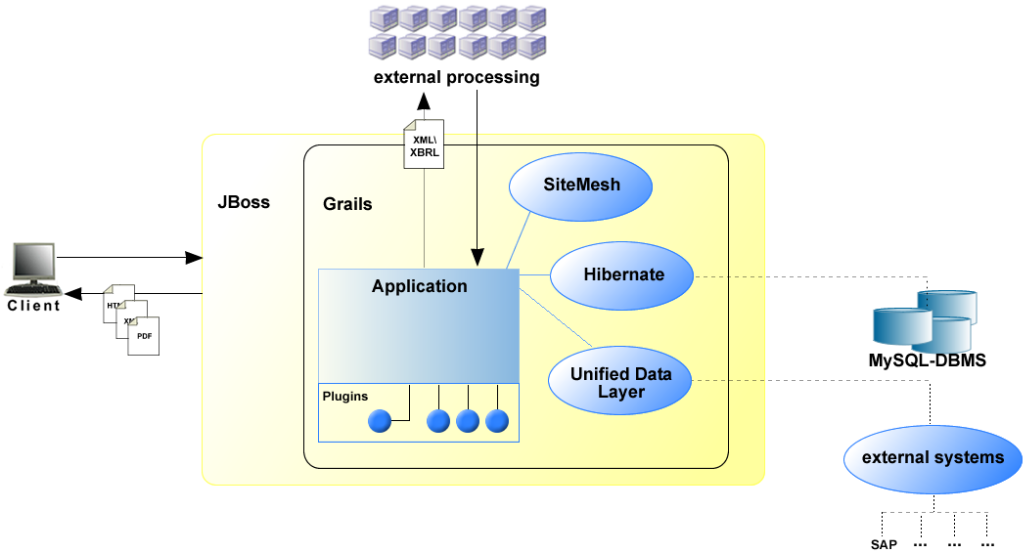

Abbildung 8.1.: Systemarchitektur

8.6. Komponenten

Nachdem nun die systemseitige Plattform beschrieben wurde, werden im Folgenden die einzelnen Softwarekomponenten genannt, die im Kapitel 9 detailliert beschrieben werden.

Die Software besteht aus folgenden Komponenten:

- Redaktionsbereich
  - Schemaeditor
  - Berichts-/Artikeleditor
- Öffentlicher Bereich
  - Web-Darstellung der Berichte/Artikel
  - Infokorb
  - Kommentarfunktion
  - Bewertungsfunktion
  - PDF-Generator
  - XML-Generator
  - RSS-Feed
- Recommender Engine
  - Datenbasis
  - Schnittstelle
  - Engine „Zuletzt angesehene Artikel“
  - Engine „Beliebteste Artikel“
  - Engine „Beste Artikel“
  - Engine „Ähnliche Artikel“
- Administrationsbereich
  - Benutzerverwaltung
  - Modulverwaltung

9. Systemkomponenten

Das vorliegende Kapitel beschreibt die Funktionalitäten des Systems, aufgeteilt in Administration, Redaktion, öffentlicher Bereich, Recommender Engine und weitere Features.

9.1. Systemadministration

In diesem Kapitel geht es um die Systemadministration. Zuerst wird dazu die Modulverwaltung erläutert. Anschließend wird die Benutzerverwaltung erörtert. Das Kapitel schließt mit der Beschreibung aller Anwendungsfälle in den Bereichen Modul- und Usermanagement ab.

9.1.1. Modulverwaltung

Dieser Abschnitt beschäftigt sich mit dem Modulkonzept. Ein Modul bildet einen bestimmten Funktionsumfang ab, wodurch die Summe aller Module die Applikation Storm bildet. Einige dieser Module können, wie in der Anforderungsdefinition in Kapitel 5.1.7 beschrieben, zusätzlich an und abgeschaltet werden. Technisch gesehen ist ein Modul ein Controller. Neben dieser Funktion besitzt jedes Modul eine Sammlung von Rechten und Parametern. Zusätzlich ermöglicht das Modulkonzept das einfache erstellen neuer Module, die automatisch die beschriebenen Meta-Information besitzen.
Zur Programmentwicklung galt es das Web-Framework Grails benutzt werden. Dieses Framework verwendet das Web MVC Konzept. Der Controller übernimmt die Steuerung von der Models und Views. Das ganze Gebilde aus Kontroller, Views und Models übernimmt eine bestimmten Aufgabenbereich und ist ein Modul. Diese Module und müssen den folgenden Anforderungen genügen:

- Einige Komponenten müssen ab- und anschaltbar sein.
- Die Komponenten müssen parametrisierbar sein.
- Die Komponenten müssen mit Rechten versehen werden.

Um die Anforderungen zu lösen, ist es nötig Informationen pro Controller zu speichern. Dieses wird gelöst indem definiert wird, dass jeder Controller eine Instance von der Klasse Modul ist. Somit können pro Model Attribute gespeichert werden. Die Instanceattribute sind Name, Description und Detachable. Das Attribut Name muss identisch mit dem Controllernamen, jedoch ohne den Zusatz Controller, sein. Dieser erfolgt aus dem Grund, damit das Modul dem Controller zugeordnet werden kann. Detachable beschreibt, ob das Modul an bzw. abgeschaltet ist. Die Rechte und Parameter sind in zwei weiteren Klassen ausgelagert. Die Klasse Config ist für die Parameter zuständig und die Klasse Rule für die Rechte. Die Abbildung 9.1

TODO BILD

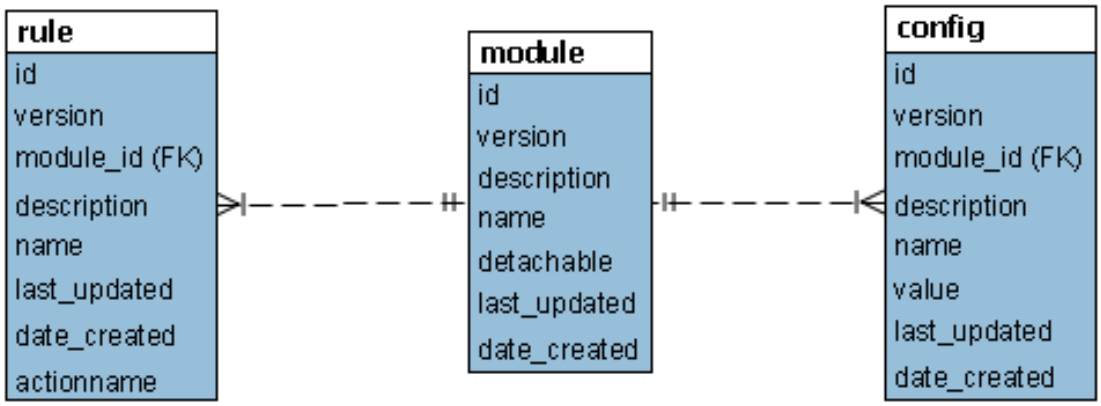

Abbildung 9.1.: ERD Modulkonzept

verdeutlich das ORM Mapping der Klassenbeziehung durch eine ERD Ansicht. Die Modulinformationen können über das Adminmenu verändert, gelöscht, angeschaut und erstellt werden. Programmtechnisch muss bei jedem Aufruf eines Controllers überprüft werden, ob dieser bereits eine Instance von der Klasse Modul hat. Dazu wird vor dem Controlleraufruf ein Grails Filter aufgerufen. Dieser Filter überprüft die Modulinformationen und legt diese an, sofern diese noch nicht vorhanden sind.

An-/Abschaltbarkeit

Die Web 2.0 Features sollen an- und abschaltbar sein, so dass jeder Betreiber der Software ihren Umfang bestimmen kann. Diese Auswahl soll dynamisch während des Betriebes geändert werden können. Die Web 2.0 Features werden programmtechnisch, wie die vorherigen Module erstellt. Somit besitzen Web 2.0 Features einen Controller und gegebenenfalls Models und Views. Im folgenden Text werden Web 2.0 Features auch als Modul angesehen. Ob ein Modul an oder abschaltbar ist, wird durch das Attribut „detachable“ festgelegt. Es kann folgende Ausprägungen annehmen:

- 0; das Modul ist deaktiviert
- 1: das Modul ist aktiv
- -1; das Modul ist aktiv und kann nicht deaktiviert werden

Mittels einem Grails Filter wird kontrolliert, bevor ein Controller aufgerufen wird, ob das Modul aktiv ist. Sollte dieses der Fall sein, wird der Controller aufgerufen, ansonsten wird der Controller übersprungen und der nächste ausgeführt. Somit kann durch das Verändern des Attributes einzelne Modul deaktviert oder aktiviert werden, ohne die gesamte Applikation zu beenden.

Erweiterbarkeit

Die Applikation soll über Programmcodeänderungen zu erweitern sein. Erweitern bezieht sich auf das Hinzufügen neuer Modul wie z.B. Web 2.0 Features. Besonders die An- und Abschaltbarkeit soll mit wenig Aufwand in Bezug auf den neuen Controller, also das neue Modul, implementierbar sein.
Ein neu entwickeltes Modul besteht aus mindestens einem Controller. Es wird bei jedem Aufruf von einem Controller kontrolliert, ob der Controller bereits einen Moduleintrag hat. Ein neuentwickelter Controller hätte keinen Eintrag. Dieser würde bei dem ersten Aufruf

automatisch, mit Default-Werten, erstellt werden. Diese Default-Werte können über den Administrationsbereich verändert werden und betreffen die Konfiguration, Rechte und die An- und Abschaltbarkeit. Nach der Erstellung der Modulinformationen verhält sich das neuentwickelte Modul wie ein Modul aus dem Auslieferungszustand. Dieses inkludiert die An- und Abschaltbarkeit, Parametrisierbarkeit und die Rechtezuteilung.

9.1.2. Rollenkonzept und Benutzerverwaltung

Die Arbeiten, die ein Administrator eines EDV-Systems erledigen muss, damit die Benutzer des von ihm betreuten Systems genau die Arbeiten erledigen können, die sie machen sollen und alles andere nicht machen können, werden unter dem Begriff Benutzerverwaltung zusammengefasst (Siehe [Unka]). In diesem Abschnitt wird beschrieben, wie die Benutzer- und Rechteverwaltung umgesetzt wird und welche Datenbasis in diesem Bereich dafür nötig ist. Grundlegend lässt sich das Rollenkonzept wie folg beschreiben: Es gibt Benutzer, Rollen und Rechte. Einem Benutzer können Rollen zugewiesen werden. Einer Rolle wiederrum können Rechte zugeordnet werden. Über die Instanz einer Rolle werden den Benutzern demnach die Rechte zugeteilt.
Den Anforderungen folgend (siehe 5.1.1) soll ein Recht einer Webseite der Software entsprechen. Aufgrund der Implementierung der Software mit dem Grails-Framework und dem damit einhergehenden MVC-Konzept entspricht eine Webseite der Software einer Controller-Action-Kombination. Jeder Controller ist dabei in der Software ebenfalls als Modul abgebildet. Jedes Recht wird einem Modul zugeordnet, damit zu jedem Recht klar ist, welchem Controller dies zugeordnet ist. Neben den Rechten pro Controller-Action-Kombination wird es pro Controller auch das Recht „Controller-*“ geben. Das Sternchen steht für die Menge aller Actions zu diesem Controller. Besitzt ein Rolle das Recht „Controller-*“ darf ein dieser Rolle zugeordneter Benutzer in diesem Controller alle Actions aufrufen und somit alle Webseiten dieses Moduls sehen. Somit ist es notwendig zu einem Recht den das entsprechende Modul, den Controllernamen und den Actionnamen zu speichern. Um die drei Stufen (User, Rolle, Recht) zu realisieren, ist es nötig die jeweiligen Stufen miteinander zu verknüpfen. Zwischen den Entitäten User und Rolle, Rolle und Recht müssen Beziehungen bestehen. Die Ausprägung dieser Beziehungen wird entsprechend den Anforderungen umgesetzt, siehe Abschnitt 5.1.1.
Für jede Controller-Action-Kombination gibt es ein Recht, Rechte werden in einer Rolle gebündelt und können so verschiedener Usern zugeordnet werden. Die zur Speicherung von Rechten, Rollen und Usern notwendigen Daten - jeweils in Form einer Datenbanktabelle - werden anhand eines ER-Diagramms in Abbildung 9.2 gezeigt.

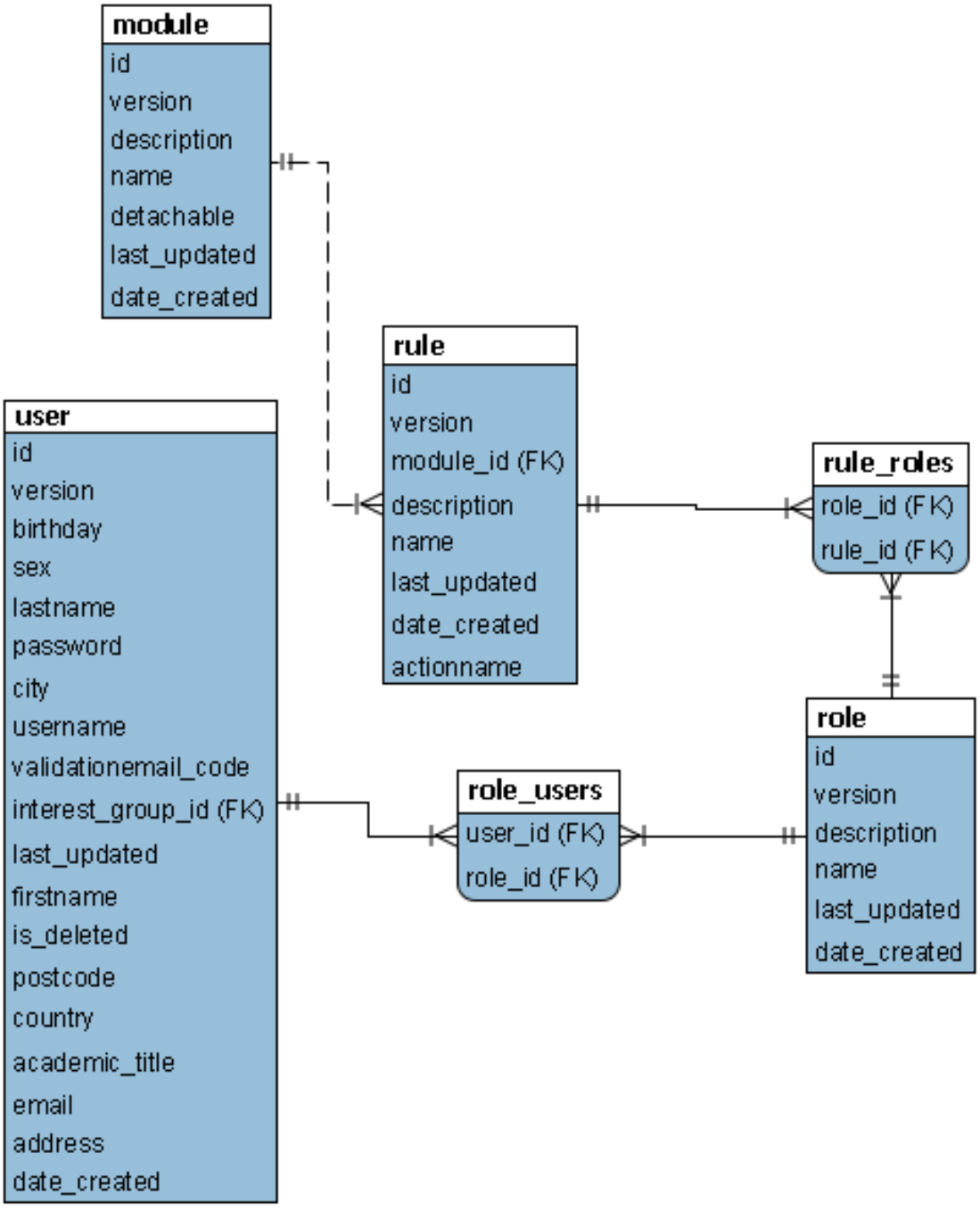

Abbildung 9.2.: ERD Rollenkonzept

Die Relation Benutzer besteht aus folgenden Pflichtfeldern: Username, Email-Adresse und Passwort. Die Felder Name, Vorname, Interessensgruppe, akademischer Titel, Geschlecht, Geburtsdatum, Adresse, Postleitzahl, Stadt und Land. Username und Passwort als Pflichtfelder dienen zum Login, die E-Mail-Adresse zur Verifikation des Users und zur Zusendung eines neuen Passworts das mittels Self-Service geändert werden kann. Das Attribut ValidationEmailCode dient zur Registrierungsbestätigung. Der Parameter Interessensgruppe ist relevant für die Vorschlagsfunktion „Artikel, die für deine Interessensgruppe interessant sind“, genaueres siehe Kapitel 9.4.6. Ein Benutzer kann andere Benutzer folgen, diese also seinen Followern hinzufügen. Dazu dient die Referenzierung der Entität User auf sich selbst (genaueres siehe Abschnitt 9.4.5). Ein weiteres Feld der Relation Benutzer ist isActive. Dies wird beim Benutzer erstellen auf true gesetzt und sobald der Benutzer sein PRofil löscht auf false. Die Felder ValidationEmailCode und isActive sieht der Benutzer selbst gar nicht.
Eine Rolle besteht aus den Parametern Name und Beschreibung sowie Feldern zur Verknüpfung mit Benutzern und Rechten.
Die Entität Recht besteht aus einem Namen und einer Beschreibung, weiterhin aus den für die Benutzerverwaltung relevanten Attributen Modul, Controllername und Actionname.

Nachdem soeben die Datenbasis im Bereich des Rollenkonzepts und der Benutzerverwaltung erläutert wurde, wird folgend auf die funktionalen Aspekte eingegangen. Um die funktionalen Aspekte zu erläutern ist es notwendig, den Status eines Benutzers zu kennen. Ein Benutzer der Software kann verschiedene Status einnehmen:

- Anonymer Benutzer
- Registrierter Benutzer

Der Prozess, der einen Wechsel dieses Status ergibt in Form einer Änderung von anonym zu registriert, ist der Prozess der Anmeldung. Für die Anmeldung ist eine Registrierung notwendig.
Diese Registrierung erfolgt unter Validierung der E-Mail-Adresse. Der anonyme Benutzer füllt nach Klick auf den Link „Registrieren“ das Registrierungsformular aus. In dieses trägt der bisher anonyme Benutzer seine Daten inklusive seiner E-Mail-Adresse ein, wählt die Zugehörigkeit zu einer Interessensgruppe aus[1] und sendet das Formular ab. An dieser Stelle wird der neue User im System angelegt und die Daten des Benutzers gespeichert. Das Passwort wird dabei mit Base64, einem einfachen Chiffrierungs-Algorithmus verschlüsselt, damit das Benutzerpasswort nicht im Klartext in der Datenbank gespeichert wird. Als nächstes wird das Feld „ValidationEmailCode“ mit einem Random-String befüllt. Es wird ein Link zur STORM-Anwendung generiert, welcher als Parameter den ValidationEmailCode enthält. An die vom Benutzer eingetragtene E-Mail-Adresse wird anschließend eine Registrierungs-Bestätigungs-Mail versand, die als Registrierungs-Bestätigungs-Link den eben generierten enthält. Der Benutzer muss zum erfolgreichen Abschließen der Registrierung auf diesen Link aus der E-Mail klicken. Beim Klick auf den Link wird der Parameter ausgelegen und mit dem Wert des Datenbankfeldes ValidationEmailCode verglichen. Stimmen diese überein, ist der User und seine E-Mail-Adresse bestätigt. Daher wird das Attribut ValidationEmailCode des zu Beginn der Registrierung angelegten Benutzers auf „1“ gesetzt. Hiermit ist der Prozess der Registrierung abgeschlossen.
Wenn sich ein aktuell anonymer User am System anmelden will, muss dieser dazu sein Benutzernamen und sein Passwort eingeben. Das Passwort wird mit dem gleichen Algorithmus wie bei der Registrierung verschlüsselt und mit dem verschlüsselten in der Datenbank

[1] Interessensgruppen können vom betreibenden Systemadministrator beliebig konfiguriert werden.

gespeicherten Passwort vergleichen. Der Username wird als String vergleichen. Wenn das Feld isActive auf true ist und die Eingabe des Benutzers (Username, Passwort) stimmen, verläuft die Anmeldung positiv und der Benutzer ist angemeldet und kann entsprechend seiner Rollen und damit entsprechend der Rechte Aktivitäten am System durchführen.
Einem registrierten Benutzer können flexibel verschiedene Rollen zugewiesen werden. Einer Rolle wiederrum können flexibel verschiedene Rechte zugewiesen werden. Durch diese Flexibilität wird gewährleistet das jede Organiastion ihre spezifischen Rollen in der Software STORM einrichten kann. Über die Struktur Benutzer-Rolle-Recht bekommt ein registrierter Benutzer ensprechende Rechte. Sobald ein Benutzer auf eine Seite der Software geht, wird ihm dieser als Benutzer „Anonym“ gesetzt. Diesen Status kann der Benutzer durch eine Anmeldung am System insofern ändern, als dass der Benutzer „Anonym“ durch seinen entsprechenden Benutzer ersetzt wird. Es wird also nie vorkommen, dass kein Sessionuser verfügbar ist.
Für jede Controller-Action-Kombination gibt es ein Recht. Somit wird der Zugriff auf jede solcher Kombinationen vom Rechtesystem verwaltet. Bei Aufruf einer Seite, sprich einer Controller-Action-Kombination, wird für den Sessionuser überprüft ob diesem über eine seiner ihm zugeordneten Rolle das entsprechende Recht verfügbar ist. Um für einen Controller alle Rechte für seine entsprechenden Action zu vergeben, gibt es das Recht „Controller-*“. Hiermit kann einer Rolle der Zugriff auf alle Actions dieses Controllers zugewiesen werden.
Der Algorithmus, der überprüft ob ein User auf eine Controller-Action-Kombination zugreifen darf, läuft wie folgt ab: Die aufzurufende Seite (Controller-Action-Kombination) ist bekannt. Zuerst wird zu dem entsprechenden User alle Rollen ermittelt. In Bezug auf diese Rollen werden die Rechte ermittelt. Die Zugriffskontrolle geschieht als Vergleich der aufzurufenden Controller-Action-Kombination mit dem den Recht anhängenden Parametern Controller und Action. Die Ermittlung der Rolle zu einem User und der Rechte zu einer Rolle erfolgt über ein direkten Zugriff auf die Datenbank - sprich mittels Hibernate-SQL - um die Performance zu optimieren. Da diese Zugriffskontrolle bei jedem Seitenaufruf ausgeführt werden muss, wird die Durchführung durch einen zentralen Filter realisiert. Dieser Filter ist entsprechend ein „befor“-Filter. Das heißt der Filter wird vor jedem Seitenzugriff ausgeführt.
Neben dieser Zugriffskontrolle auf Seitenebene kann es den Fall geben, dass ein User eine Seite betrachten darf, aber bestimmte Inhalte dieser Seite nicht. Beispielsweise darf ein Redaktuer einen Nachhaltigkeitsbericht sehen und bearbeiten, diesen Nachhaltigkeitsbericht aber nicht freigeben. Auf der entsprechenden Seite muss demnach der Button „Freigeben“ versteckt werden, wenn keiner Rolle des User das Recht „Nachhaltigkeitsbericht freigeben“ zugeordnet ist. Diese Anforderung wird durch ein Tag realisiert. Bei der Einbindung des Buttons auf die Seite wird diesem Code ein Tag vorgeschaltet, dass als Input-Parameter den aufzurufenden Controller und die aufzurufende Action bekommt (z.B. Nachhaltigkeitsbericht - Freigeben) und als Rückgabe im Falle einer positiven Rechteprüfung den Code des Buttons und im negativen Fall nichts liefert.

Die Pflege der Beziehungen zwischen User und Rolle sowie zwischen Rolle und Recht erfolgt über die zentrale Komponente Rolle. Einer Rolle können mehrere Rechte und mehrere User zugewiesen werden. Im Auslieferungszustand wird es vier Rollen geben:

- Anonym: Anonymer User. Diese Rolle ist dem anonymen User zugeteilt. Den User-Status „Anonym“ bekommt jeder Webseitenbesucher initial zugewiesen. Diese Rolle kann beispielsweise den Info-Korb benutzen, Artikel betrachten und bekommt die Vorschläge angezeigt, die nicht als Input userspezifische Daten (wie z.B. die Interes-

sengruppe) benötigen. (Kann ein Anonymer auch bewerten/kommentieren?)

- User: Registrierter User. Diese Rolle kann beliebige Info-Korb-Zusammenstellung abspeichern, bekommt userspezifische Vorschläge und kann Bewertungen und Kommentierungen vornehmen.
- Redaktuer: Ein Redaktuer kann einen Nachhaltigkeitsbericht erstellen, den Content eines Artikels pflegen und die Mediathek verwalten. Weiterhin kann diese Rolle einen Nachhaltigkeitsbericht freigeben und verschiedene Schemata anlegen und verwalten.
- Administratr: Die Rolle Administrator besitzt alle Rechte. Ein User mit dieser Rolle darf beispielsweise die User-Rolle-Rechte-Zuordnung ändern, Module an- und abschalten, die Konfiguration von Modulen ändern.

Im Auslieferungszustand ist sind diese Rollen Rechte vergeben. Dieser Auslieferungszustand ist absichtlich auf die grundlegenden Rollen beschränkt. Das Rollenkonzept mit der Struktur User-Rolle-Recht bietet jeder die Software STORM einsetzenden Organisation eine hohe Flexibilität. Individuelle Organisationsstrukturen können somit abgebildet werden. Durch diese Adaptivität kann STORM kann in Bezug auf die Benutzerverwaltung flexibel eingesetzt werden. Der Auslieferungszustand der Rechtekonfiguration kann detailliert im Anhang D aufgeführt.
Als weitere Funktionalität ist es möglich, dass ein User einem anderen folgt - diesen also als Follower hinzufügt. Damit kann der User andere für ihn interessante User verfolgen. Diese Verfolgung wird für das Vorschlagswesen (Siehe Kapitel 9.4.5) insofern relevant, als dass es möglich sein wird, bestimmte Follower hinzuzufügen und deren betrachtete Artikel als Vorschläge aus seinem eigenen Netzwerk zu bekommen. Ein Szenario in einer Organisation könnte sein, dass ein Student seinen Professor als Follower hinzufügt um somit die aus professoraler Sicht interessanten Artikel ebenfalls zu lesen.

9.1.3. Anwendungsfälle im Bereich Systemadministration

In diesem Abschnitt werden die Anwendungsfälle, die in der Anforderungsdefinition indentifiziert wurden (siehe 5.1.3), konkretisiert. Zunächst werden die Use Cases vorgestellt die ein registrierter Benutzer dürfen kann. Anschließend wird auf Use Cases eingegangen, die nur ein Benutzer mit entsprechenden Rechten durchführen kann.

Registrieren

Ein neuer Benutzer kann sich über den Link „Registrieren“ in der Menge aller Benutzer aufnehmen lassen. In dem Registrierungsprozess füllt der neue Benutzer zunächst die Felder der Registrierungsmaske aus. Nach dem Absenden dieses Formulars wird dem neuen Benutzer eine E-Mail mit Registrierungs-Bestätigungs-Link zugestellt. Durch Klick auf den Link wird der Prozess der Registrierung positiv beendet. Erfolg kein Klick auf diesen Link, wird der Benutzer nicht für die Anmeldung am System freigegeben. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe 9.1.

Name	Registrieren
Beschreibung	Ein neuer Benutzer registriert sich am System
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Registrieren“
Vorbedingungen	Benutzer ist aktuell nicht angemeldet.

Nachbedingung/Ergebnis	Benutzer ist registriert und bekommt eine E-Mail mit Registrierungs-Bestätigungs-Link. Nachdem der Benutzer auf diesen Link geklickt hat, seine E-Mail-Adresse somit also bestätigt ist, ist der Benutzer für eine Anmeldung freigegeben und die Registrierung damit abgeschlossen.

Tabelle 9.1.: Anwendungsfall: „Registrieren“

Anmelden

Ein bisher anonyme Benutzer vollzieht den Prozess der Anmeldung. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe 9.2.

Name	Anmelden
Beschreibung	Ein Benutzer meldet sich am System an.
Beteiligte Akteure	Benutzer
Auslöser	Ein Benutzer klickt auf „Anmelden“.
Vorbedingungen	Benutzer ist nicht angemeldet.
Nachbedingung/Ergebnis	Wurde das Passwort und der Benutzername korrekt eingegeben, erfolgt der Login und die Startseite wird angezeigt.

Tabelle 9.2.: Anwendungsfall: „Anmelden“

Profil ansehen

Jeder angemeldete Benutzer kann sein eigenes Profil ansehen. Mehr Informationen zu diesem Use Case können der Tabelle 9.3 entnommen werden.

Name	Profil ansehen
Beschreibung	Die vom Benutzer im System hinterlegten Daten können betrachtet werden.
Beteiligte Akteure	Benutzer
Auslöser	Ein Benutzer klickt auf „Profil verwalten“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Änderungen an den Profildaten des Benutzers sind gespeichert.

Tabelle 9.3.: Anwendungsfall: „Profil ansehen“

Profil ändern

Jeder angemeldete Benutzer kann sein eigenes Profil ändern. Es können Änderungen vorgenommen und diese gespeichtert werden. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe 9.4.

Name	Profil ändern

Beschreibung	Benutzerspezifische Daten können geändert und gespeichert werden.
Beteiligte Akteure	Benutzer
Auslöser	Ein Benutzer klickt auf „Profil ändern“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Änderungen an den Profildaten des Benutzers sind gespeichert.

Tabelle 9.4.: Anwendungsfall: „Profil ändern“

Profil löschen

Jeder angemeldete Benutzer kann sein eigenes Profil löschen und somit seinen Zugangsaccount entfernen. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe 9.5.

Name	Profil löschen
Beschreibung	Der Benutzer entfernen seinen User und somit sein Zugang zu dem System.
Beteiligte Akteure	Benutzer
Auslöser	Ein Benutzer klickt auf „Profil löschen“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	User wird aus der DB gelöscht.

Tabelle 9.5.: Anwendungsfall: „Profil löschen“

Startseite betrachten

Der Benutzer betrachtet die Startseite. Diese wird nach erfolgreichem Login oder bei Klick auf das STORM Logo angezeigt. Mehr Informationen zu diesem Use Case können der Tabelle 9.6 entnommen werden.

Name	Startseite betrachten
Beschreibung	Der Benutzer entfernen seinen User und somit sein Zugang zu dem System.
Beteiligte Akteure	Benutzer
Auslöser	Der Benutzer meldet sich an, danach wird die Startseite aufgerufen. Die Startseite kann auch über ein Klick auf das STORM Logo erreicht werden.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Die Startseite wird angezeigt.

Tabelle 9.6.: Anwendungsfall: „Startseite betrachten“

Meine Followers ansehen

Ein Benutzer kann sich die Liste aller Follower, also der User denen er aktuell folgt, ansehen. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe 9.7.

Name	Meine Followers ansehen
Beschreibung	Der Benutzer betrachtet die Lister aller User denen der Benutzer aktuell folgt.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Meine Follower“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Alle Follower werden aufgelistet.

Tabelle 9.7.: Anwendungsfall: „Meine Followers ansehen“

Followers hinzufügen

Ein Benutzer kann andere Benutzer zu seinen Followern hinzufügen. Dazu geht dieser Benutzer auf das Profil desjenigem dem gefolgt werden soll und fügt ihn dort seiner Liste hinzu. Diese Benutzer werden fortan im dem Vorschlagswesen „Artikel, die deine Follower betrachtet haben“ (siehe Abschnitt 9.4.5) berücksichtigt. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.8.

Name	Followers hinzufügen
Beschreibung	Der Benutzer möchte einen anderen Benutzer zu der Liste seiner Follower hinzufügen.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt in einer Ansicht eines anderen Users auf das Symbol „Als Follower hinzufügen“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Alle Follower werden aufgelistet.

Tabelle 9.8.: Anwendungsfall: „Followers hinzufügen“

Followers entfernen

Ein Benutzer andere Benutzer als der Liste seiner Follower entfernen. Dazu geht dieser Benutzer auf die Liste aller Follower und entfernt dort den entsprechenden Follower, genaueres siehe Tabelle 9.9.

Name	Followers entfernen
Beschreibung	Der Benutzer möchte einen seiner Follwer als Follower entfernen.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt in der Ansicht seiner Follower in Bezug auf deinen Follower auf „Als Follower entfernen“.
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Alle Follower ohne den gerade Entfernten werden aufgelistet.

Tabelle 9.9.: Anwendungsfall: „Follower entfernen“

Eigenes Passwort ändern

Ein Benutzer möchte sein eigenes Passwort ändern. Mehr Informationen zu diesem Use Case können der Tabelle 9.10 entnommen werden.

Name	Eigenes Passwort ändern
Beschreibung	Ein Benutzer ändert sein Passwort.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Passwort ändern“
Vorbedingungen	Benutzer ist angemeldet.
Nachbedingung/Ergebnis	Neues Passwort ist gesetzt.

Tabelle 9.10.: Anwendungsfall: „Eigenes Passwort ändern“

Passwort zurücksetzen

Ein Benutzer hat sein Passwort vergessen und ändert dies nun über sie Self-Service-Komponente. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.11.

Name	Passwort zurücksetzen
Beschreibung	Ein Benutzer setze sein Passwort zurück.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Passwort vergessen?“
Vorbedingungen	Benutzer ist nicht angemeldet.
Nachbedingung/Ergebnis	Neues Passwort wird per E-Mail zugeschickt.

Tabelle 9.11.: Anwendungsfall: „Passwort zurücksetzen “

Protokollierung ansehen

Ein Benutzer mit entsprechenden Rechten sieht sich den aktuellen Stand der Protokollierung an. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.12.

Name	Protokollierung ansehen
Beschreibung	Ein Benutzer schaut sich die Protokollierung an.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Protokollierung anzeigen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Protokollierung wird angezeigt.

Tabelle 9.12.: Anwendungsfall: „Protokollierung ansehen“

Recht ansehen

Ein Benutzer mit entsprechenden Rechten sieht sich ein Recht an. Mehr Informationen zu diesem Use Case können der Tabelle 9.13 entnommen werden.

Name	Recht ansehen

Beschreibung	Ein Benutzer schaut sich ein Recht an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Recht anzeigen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Recht wird angezeigt.

Tabelle 9.13.: Anwendungsfall: „Recht ansehen“

Neues Recht hinzufügen

Ein Benutzer mit entsprechenden Rechten erstellt ein neues Recht. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.14.

Name	Neues Recht hinzufügen
Beschreibung	Ein Benutzer legt ein neues Recht an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Recht anlegen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Recht wird gespeichert und angezeigt.

Tabelle 9.14.: Anwendungsfall: „Neues Recht hinzufügen“

Recht ändern

Ein Benutzer mit entsprechenden Rechten ändert ein Recht. Mehr Informationen zu diesem Use Case können der Tabelle 9.15 entnommen werden.

Name	Recht ändern
Beschreibung	Ein Benutzer ändert ein Recht.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Recht ändern“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Änderungen am Recht sind gespeichert und werden angezeigt.

Tabelle 9.15.: Anwendungsfall: „Recht ändern“

Recht löschen

Ein Benutzer mit entsprechenden Rechten löscht ein Recht. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.16.

Name	Recht löschen
Beschreibung	Ein Benutzer löscht ein Recht.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten

Auslöser	Benutzer klickt auf „Recht löschen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen. Das Recht ist keiner Rolle zugeordnet.
Nachbedingung/Ergebnis	Das Recht ist gelöscht.

Tabelle 9.16.: Anwendungsfall: „Recht löschen“

Rolle ansehen

Ein Benutzer mit entsprechenden Rechten sieht sich eine Rolle an. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.17.

Name	Rolle ansehen
Beschreibung	Ein Benutzer schaut sich eine Rolle an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Rolle anzeigen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Rolle wird angezeigt.

Tabelle 9.17.: Anwendungsfall: „Rolle ansehen“

Neue Rolle hinzufügen

Ein Benutzer mit entsprechenden Rechten erstellt eine neue Rolle. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabélle 9.18.

Name	Neue Rolle hinzufügen
Beschreibung	Ein Benutzer legt eine neue Rolle an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Rolle anlegen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Rolle wird gespeichert und angezeigt.

Tabelle 9.18.: Anwendungsfall: „Neue Rolle hinzufügen“

Rolle ändern

Ein Benutzer mit entsprechenden Rechten ändert eine Rolle. Mehr Informationen zu diesem Use Case können der Tabelle 9.19 entnommen werden.

Name	Rolle ändern
Beschreibung	Ein Benutzer ändert eine Rolle.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Rolle ändern“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.

Nachbedingung/Ergebnis	Änderungen an der Rolle sind gespeichert und werden angezeigt.

Tabelle 9.19.: Anwendungsfall: „Rolle ändern“

Recht löschen

Ein Benutzer mit entsprechenden Rechten löscht eine Rolle. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.20.

Name	Rolle löschen
Beschreibung	Ein Benutzer löscht eine Rolle.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Rolle löschen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen. Der Rolle ist kein User zugeordnet.
Nachbedingung/Ergebnis	Die Rolle ist gelöscht.

Tabelle 9.20.: Anwendungsfall: „Rolle löschen“

Passwort ändern

Ein Benutzer mit entsprechenden Rechten ändert das Passwort eines anderen Benutzers. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.21.

Name	Passwort ändern
Beschreibung	Ein Benutzer ändert das Passwort eines anderen Benutzers.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Passwort ändern“
Vorbedingungen	Benutzer ist angemeldet und besitzt entsprechende Rechte.
Nachbedingung/Ergebnis	Neues Passwort ist gesetzt.

Tabelle 9.21.: Anwendungsfall: „Passwort ändern“

Benutzer betrachten

Ein Benutzer betrachtet das Profil eines anderen Benutzers. Mehr Informationen zu diesem Use Case können der Tabelle 9.22 entnommen werden.

Name	Benutzer betrachten
Beschreibung	Es wird das Profil eines Benutzers betrachtet.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „User anzeigen“
Vorbedingungen	Benutzer ist angemeldet und besitzt das entsprechende Recht.

Nachbedingung/Ergebnis	Benutzerprofil wird angezeigt.

Tabelle 9.22.: Anwendungsfall: „Benutzer betrachten“

Neuen Benutzer anlegen

Ein Benutzer mit entsprechenden Rechten legt einen neuen Benutzer im System an. Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.23.

Name	Neuen Benutzer anlegen
Beschreibung	Es wird das Profil eines Benutzers betrachtet.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Neuen Benutzer anlegen.“
Vorbedingungen	Benutzer ist angemeldet und besitzt das entsprechende Recht.
Nachbedingung/Ergebnis	Benutzer ist angelegt und wird angezeigt.

Tabelle 9.23.: Anwendungsfall: „Neuen Benutzer anlegen“

Benutzer ändern

Ein Benutzer ändert das Profil eines anderen Benutzers. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.24.

Name	Benutzer ändern
Beschreibung	Es wird das Profil eines Benutzers geändert.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Benutzer ändern“
Vorbedingungen	Benutzer ist angemeldet und besitzt das entsprechende Recht.
Nachbedingung/Ergebnis	Benutzerprofil wird geändert und angezeigt.

Tabelle 9.24.: Anwendungsfall: „Benutzer ändern“

Benutzer löschen

Ein Benutzer löscht das Profil eines anderen Benutzers. Mehr Informationen zu diesem Use Case können der Tabelle 9.25 entnommen werden.

Name	Benutzer löschen
Beschreibung	Es wird das Profil eines Benutzers geändert.
Beteiligte Akteure	Benutzer
Auslöser	Benutzer klickt auf „Benutzer löschen“
Vorbedingungen	Benutzer ist angemeldet und besitzt das entsprechende Recht.
Nachbedingung/Ergebnis	Benutzerprofil wird gelöscht.

Tabelle 9.25.: Anwendungsfall: „Benutzer löschen“

Konfiguration betrachten

Ein Benutzer mit entsprechenden Rechten sieht sich eine Konfiguration an. Mehr Informationen zu diesem Use Case können der Tabelle 9.26 entnommen werden.

Name	Konfiguration betrachten
Beschreibung	Ein Benutzer schaut sich eine Konfiguration für ein Modul an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Konfiguration betrachten“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	‘Konfiguration wird angezeigt.

Tabelle 9.26.: Anwendungsfall: „Konfiguration betrachten“

Neue Konfiguration anlegen

Ein Benutzer mit entsprechenden Rechten erstellt eine neue Konfiguration für ein Module. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle 9.27.

Name	Neue Konfiguration anlegen
Beschreibung	Ein Benutzer legt eine neue Konfiguration für ein Module an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Konfiguration anlegen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Konfiguration wird gespeichert und angezeigt.

Tabelle 9.27.: Anwendungsfall: „Neue Konfiguration anlegen“

Konfiguration löschen

Ein Benutzer mit entsprechenden Rechten ändert eine Konfiguration. Mehr Informationen zu diesem Use Case können der Tabelle 9.28 entnommen werden.

Name	Konfiguration löschen
Beschreibung	Ein Benutzer löscht eine ‘Konfiguration.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Konfiguration löschen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.Die Konfiguration ist keinem Module zugeordnet
Nachbedingung/Ergebnis	Konfiguration wurde gelöscht und die Liste aller Konfiguration wird angezeigt.

Tabelle 9.28.: Anwendungsfall: „Konfiguration löschen“

Konfiguration ändern

Ein Benutzer mit entsprechenden Rechten ändert eine Konfiguration . Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.29.

Name	Konfiguration ändern
Beschreibung	Ein Benutzer ändert eine Konfiguration.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Konfiguration ändern“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen. .
Nachbedingung/Ergebnis	Die Konfiguration wurde geändert und wird angezeigt.

Tabelle 9.29.: Anwendungsfall: „Konfiguration ändern“

Modul betrachten

Ein Benutzer mit entsprechenden Rechten sieht sich ein Modul an. Mehr Informationen zu diesem Use Case können der Tabelle 9.30 entnommen werden.

Name	Konfiguration betrachten
Beschreibung	Ein Benutzer schaut sich ein Modul an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Modul betrachten“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Modul wird angezeigt.

Tabelle 9.30.: Anwendungsfall: „Modul betrachten“

Neues Modul anlegen

Ein Benutzer mit entsprechenden Rechten erstellt eine neues Module. Nachfolgende Tabelle konkretisiert diesen Anwendungsfall, siehe Tabelle **??**.

Name	Neues Module anlegen
Beschreibung	Ein Benutzer legt eine neues Module an.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Neues Modul anlegen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen.
Nachbedingung/Ergebnis	Modul wird gespeichert und angezeigt.

Tabelle 9.31.: Anwendungsfall: „Neues Modul anlegen“

Modul löschen

Ein Benutzer mit entsprechenden Rechten löscht ein Modul. Mehr Informationen zu diesem Use Case können der Tabelle 9.32 entnommen werden.

Name	Modul löschen
Beschreibung	Ein Benutzer löscht ein Modul.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Modul löschen“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen. Das Modul hat keine zugeordneten Rechte oder Konfiguration
Nachbedingung/Ergebnis	Modul wurde gelöscht und die Liste aller Konfiguration wird angezeigt.

Tabelle 9.32.: Anwendungsfall: „Modul löschen“

Modul ändern

Ein Benutzer mit entsprechenden Rechten ändert ein Modul . Folgende Tabelle gibt einen detaillierteren Überblick über diesen Use Case, siehe Tabelle 9.33.

Name	Modul ändern
Beschreibung	Ein Benutzer ändert ein Modul.
Beteiligte Akteure	Benutzer mit entsprechenden Rechten
Auslöser	Benutzer klickt auf „Modul ändern“
Vorbedingungen	Benutzer muss angemeldet sein und über entsprechende Rechte verfügen. .
Nachbedingung/Ergebnis	Das Modul wurde geändert und wird angezeigt.

Tabelle 9.33.: Anwendungsfall: „Modul ändern“

9.2. Redaktionsbereich

Der Redaktionsbereich ist aufgeteilt in den Schemaeditor (Erstellung und Verwaltung von Schemata) und den Berichts-/Artikeleditor (auch Eingabeeditor genannt; zur Erstellung und Verwaltung von NBs und Artikeln).

9.2.1. Schemaeditor

In der Nachhaltigkeitsberichterstattung hat sich der GRI-Berichtsrahmen als Standard etabliert. Fachanwender, die den Schemaeditor nutzen (Schemaverwalter genannt) sollen in die Lage versetzt werden, Vorgaben der GRI informationstechnisch zu erfassen und in einem Schema speichern zu können. Im Folgenden wird der GRI-Berichtsrahmen genauer untersucht, um ein praxistaugliches Schema-Modell herzuleiten.

Der GRI-Berichtsrahmen

Der GRI-Berichtsrahmen dient als Leitwerkzeug bei der Erstellung von Nachhaltigkeitsberichten und unterteilt sich in vier wesentliche Bereiche:

- G3-Leitfaden
- Anwendungsebenen

- Indikatorprotokollsätze
- Technische Protokollsätze

Für die Herleitung des Schema-Modells sind die Anwendungsebenen und Indikatorprotokollsätze von besonderem Interesse, da sie die Charakteristika, der für ein Schema notwendigen Informationen beschreiben. Eine vollstänige Beschreibung aller vier Bereiche ist unter (SEmSeb) zu finden.

Anwendungsebenen

Die Anwendungsebenen dienen der Einordnung eines Nachhaltigkeitsberichts nach GRI, um kenntlich zu machen, wie vollständig der GRI-Berichtsrahmen angewendet wurde. Die GRI bietet zu diesem Zweck drei Ebenen (A,B,C). Die Einstufung des Nachhaltigkeitsberichts anhand einer dieser drei Ebenen, leistet der Ersteller des Berichts. Ein im Schemaeditor erstelltes Schema soll diese Einstufung bereits vor Erstellung des NBs erlauben, indem alle Berichtsinhalte der zu erreichenden Anwendungsebene in diesem Schema definiert sind. Eine Unterteilung der Berichtsinhalte in *pflicht* und *optionale* Berichtsinhalte ermöglicht es, den niedrigsten, aber dennoch GRI-validen Detailgrad (siehe Abbildung 9.3) über Pflichtinhalte und alle Anwendungsebenen über optionale Berichtsinhalte abzubilden.

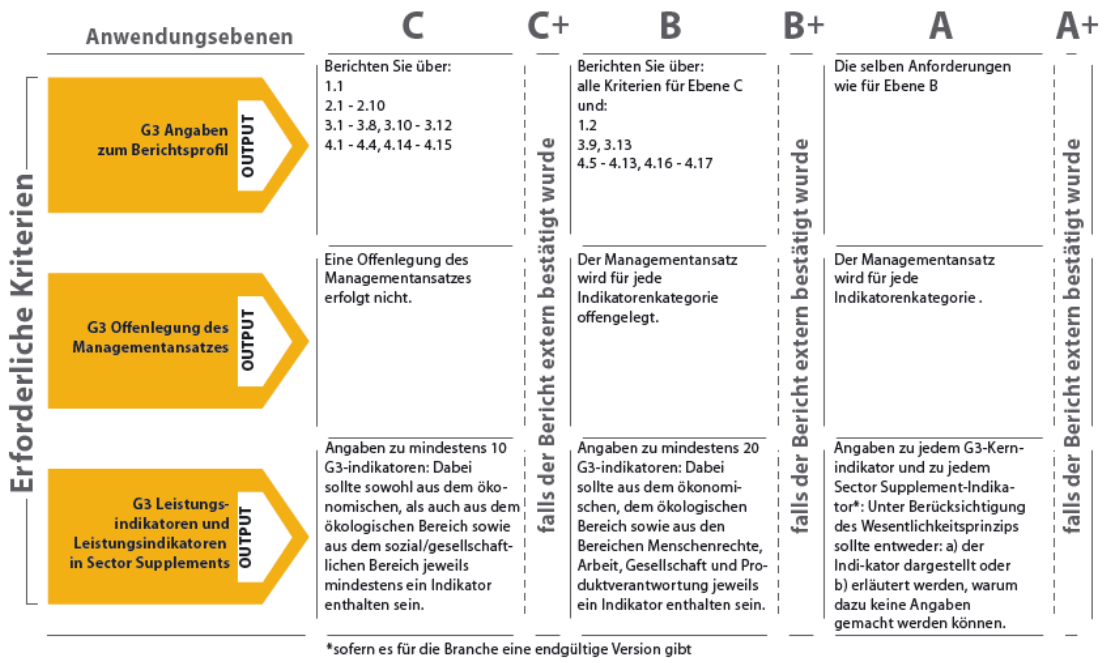

Abbildung 9.3.: Anwendungsebenen nach GRI

Indikatorenprotokollsätze der GRI

Die Indikatorprotokollsätze beinhalten alle Leistungsindikatoren nach GRI und spiegeln die drei Säulen der Nachhaltigkeit wieder[2]

9.2.2. Eingabeeditor für Berichte und Artikel

Dieses Kapitel beschreibt den Eingabeeditor für das Verwalten von konkreten NBs und Artikeln. Bezogen auf das ER-Diagramm (siehe Abbildung **??**) sind dies die Entitäten Ebene, Artikel und Report. Änderungen an der Struktur des konkreten Nachhaltigkeitsberichtes (z.B. Ebene hinzufügen/löschen, Indikator hinzufügen/ändern/löschen) werden hier nicht beschrieben, da diese Funktionalitäten analog zum Schemaeditor durchführbar sind.

Nachhaltigkeitsberichte verwalten

Es folgt die Beschreibung zum Erstellen und Verwalten von Nachhaltigkeitsberichten. Dazu wird auf das *Datenmodell*, auf den Anwendungsfall *Neuen NB erstellen* und die dazugehörige *Oberfläche* eingegangen, sowie die möglichen Status eines NBs beschrieben.
Das *Datenmodell* für einen konkreten NB (auf dem die nachfolgenden Anwendungsfälle basieren) ist wie folgt aufgebaut:

id	Eindeutige ID eines konkreten NBs. Autoincrement. Primärschlüssel.
name	Frei wählbarer Name des NBs
date	Timestamp, wann der NB erstellt wurde.
description	Frei eingebbare Beschreibung des NBs.
status	Mögliche Status eines NBs: • „Neu“ • „In Bearbeitung“ • „Geschlossen“ • „Veröffentlicht“
author_id	ID des Autors, der diesen NB erstellt hat.
schema_id	ID des Schemas, auf dem dieser NB aufbaut.
language_id	ID der Sprache, in der der komplette NB verfasst ist.

Tabelle 9.34.: Datenbanktabelle „Report“

Das Erstellen von NBs kann unter dem Punkt „Neuen NB erstellen“ im Bereich Eingabeeditor durchgeführt werden. Das Verwalten bzw. Bearbeiten von bereits vorhandenen NBs ist unter dem Punkt *NB verwalten* im Bereich Eingabeeditor zu finden. Zuerst wird der Anwendungsfall *Neuen NB erstellen* betrachtet.

Zum Erstellen eines NBs wird ein Schema benötigt, das vorher definiert worden sein muss. Aus einer Liste aller verfügbaren Schemata muss eines ausgewählt werden, auf dem der neue NB basieren soll. Danach wird das komplette Schema (also die Datenstruktur) in den neuen NB kopiert. Soll das Schema des erstellten NBs geändert werden, so ist dies über die Verwaltung von NBs möglich. Nachträgliche Änderungen am Schema haben keine Auswirkungen auf den neu erstellten NB (siehe Informationen zu Schema-Verwaltung im Bereich Schema).

Name	NB anlegen
Beschreibung	Der Benutzer hat die Möglichkeit, einen neuen NB zu erstellen.
Beteiligte Akteure	Client, Server
Auslöser	Der Benutzer möchte einen neuen NB anlegen.
Vorbedingungen	Es wird beim Usermanagement die Berechtigung für diese Aktion für den jeweiligen User abgefragt. Nur wenn eine entsprechende Berechtigung vorliegt, kann der Prozess fortgesetzt werden. Ohne Schema kann kein neuer NB angelegt werden, d.h. es muss zuerst mindestenst ein Schema als Vorlage angelegt worden sein, auf dem der NB basieren soll.
Nachbedingung/Ergebnis	Der Benutzer hat erfolgreich einen neuen NB angelegt. Dem NB wurde der Status „Neu“ zugewiesen.
Standardablauf	siehe Abbildung 9.4

Tabelle 9.35.: Anwendungsfall „NB anlegen“

Die *Oberfläche* zum Anlegen eines NBs wird in Abbildung 9.5 beispielhaft dargestellt.

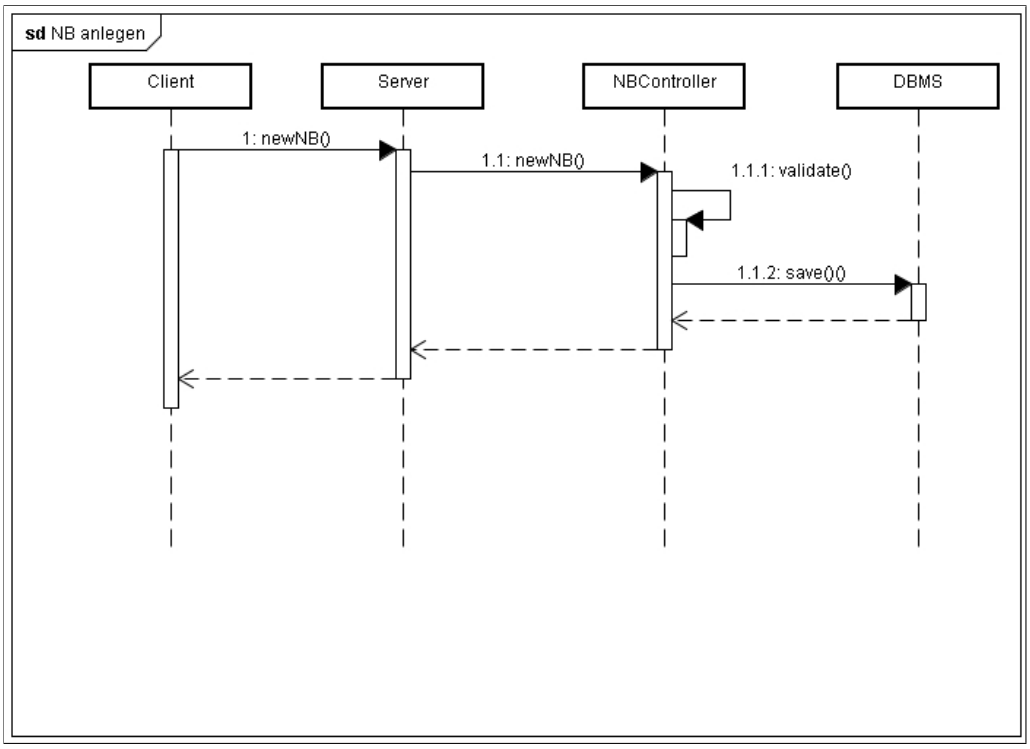

Abbildung 9.4.: Sequenzdiagramm „NB anlegen“

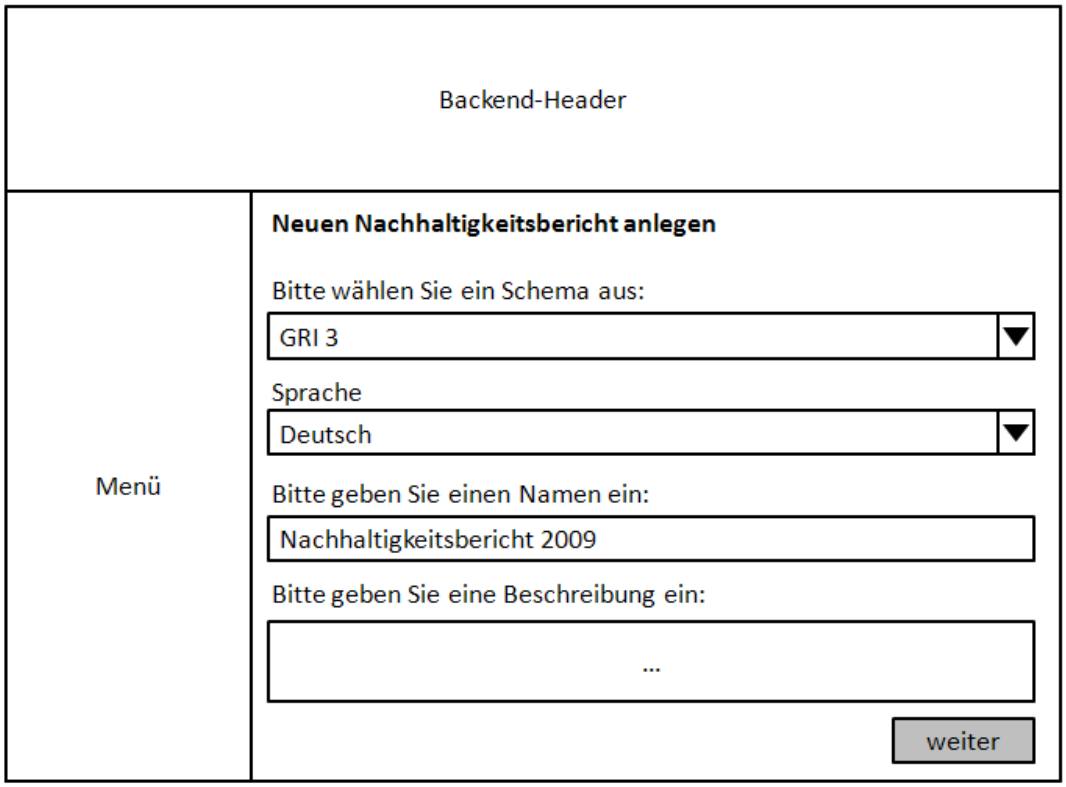

Abbildung 9.5.: Oberflächenentwurf „Neuen NB anlegen“

Der neu erstellte NB bekommt automatisch den Status *Neu* und es kann nun mit der Verwaltung der einzelnen Artikel für diesen NB begonnnen werden (siehe 9.2.2). Diese Artikel werden beim Anlegen eines neuen NBs auf Basis eines Schemas aus den vorgegebenen Schema-Ebenen generiert. Mit dem Beginn der Artikelverwaltung wird der Status des NBs automatisch *in Bearbeitung* gesetzt. (siehe 9.2.2). Solange ein NB diesen Status hat, dürfen Änderungen an diesem NB vorgenommen werden. Alle Nachhhaltigkeitsberichte können in der Übersicht *Nachhhaltigkeitsberichte* zentral dargestellt werden. Von hier aus ist die weitere Bearbeitung eines NBs möglich. Der Zusammenhang der Anwendungsfälle wir aus der Navihationshierarchie deutlich 9.6.

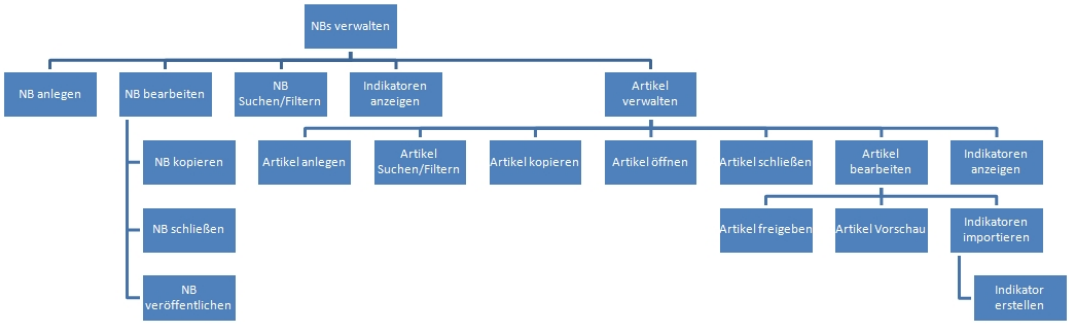

Abbildung 9.6.: Navigationshierarchie für den Eingabeeditor

Zu den weiteren Aktivitäten gehören: NB bearbeiten, Liste der Artikel oder Liste der Indikatoren des NBs anschauen. Auf der Seite *Liste der Artikel* ist die weitere Bearbeitung der Artikel gewährleistet (siehe *Artikel verwalten*) und ein Klick auf *Liste der Indikatoren* liefert eine Liste der zugehörigen Indikatoren. Wenn die Beschreibung oder den Status des NBs geändert werden soll, ist die Funktion *NB bearbeiten* zu verwenden. Benutzer können in der Übersicht nach NBs *Suchen* und *Filtern*. Neben der Freitextsuche sind folgende Filter-Kriterien möglich:

- Schema
- Status
- Autor

Ein NB kann u.a. auch den Status *Veröffentlicht* oder *Geschlossen* haben. Den Status „Veröffentlicht“ bekommt ein NB, wenn alle Artikel dieses NBs den Status „Freigegeben“ haben. Der Status „Veröffentlicht“ bedeutet, dass der NB auf der Webseite veröffentlicht wird. Dieser kann nur von einer verantwortlichen Person mit der entsprechenden Berechtigung vergeben werden. Ein NB kann auch gelöscht werden. Dabei wird der der NB nicht aus dem Datenbank-System entfernt, sondern nur sein Status auf *Geschlossen* gesezt. Der Status *Geschlossen* gibt an, dass der NB für die Bearbeitung gesperrt wurde und nicht mehr auf der Webseite angezeigt wird. Weder Artikel, noch Name, noch die Beschreibung können jetzt noch verändert werden. Ist eine Änderung jedoch nötig, so kann der Status "Geschlossen"wieder aufgehoben und der Status in *in Bearbeitung* geändert werden. Diese Rückführung kann nur von einer Person mit der entsprechenden Berechtigung durchgeführt werden (Gleiches gilt für den Status „Veröffentlicht“).

Artikel verwalten

Alle Artikel eines NBs können in der *Artikel-Übersicht* zentral dargestellt werden. Von hier aus ist die weitere Bearbeitung eines Artikels möglich. Die Oberfläche für die Übersicht ist in Abbildung 9.7 beispielhaft dargestellt.

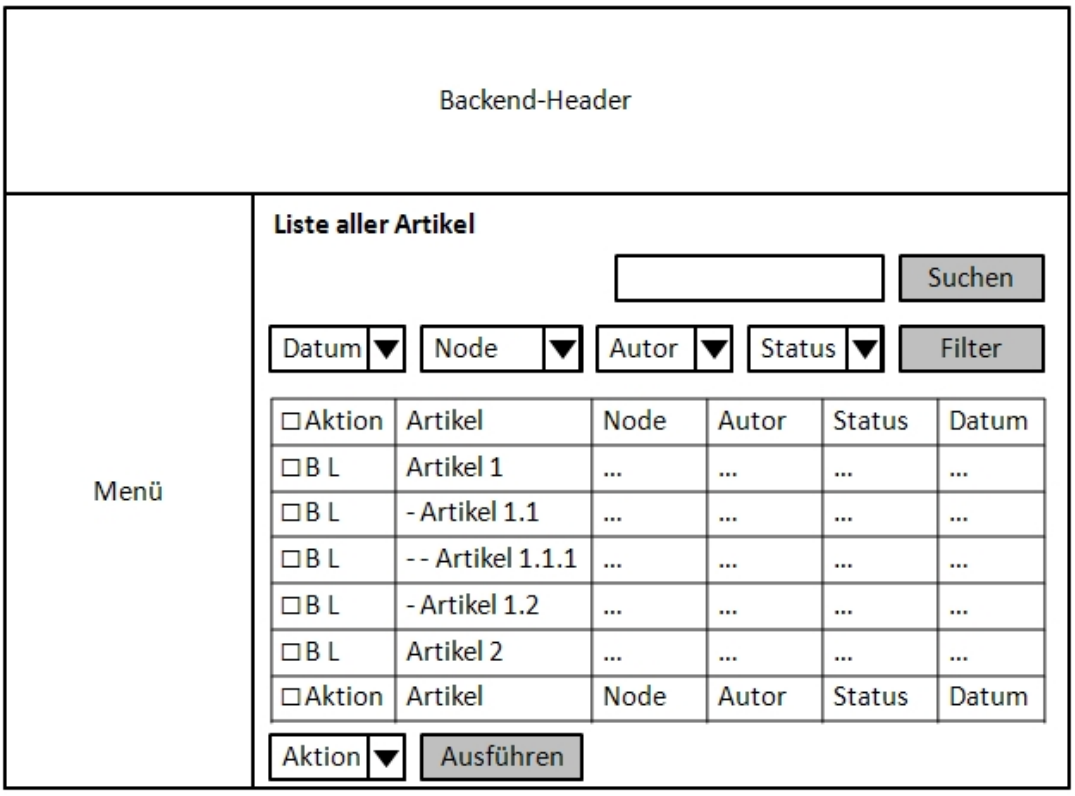

Abbildung 9.7.: Oberflächenentwurf „Artikel Übersicht“

Benutzer können in der Übersicht nach Artikeln *Suchen* und *Filtern* . Neben der Freitextsuche sind folgende Filter-Kriterien möglich:

- Status
- Autor

Die Suche beinhaltet außerdem eine Autovervollständigung, sodass dem Benutzer bereits beim Eingeben des Suchwortes mögliche Ergebnissen angeboten werden. Falls der Benutzer sich vertippen sollte, werden ihm ähnliche Suchbegriffe angezeigt, die gefunden werden können. Weitere Aktivitäten sind in dieser Übersicht möglich: Artikel bearbeiten, Vorschau, Artikel kopieren, schließen (löschen) und öffnen. Unter das Verwalten von Artikeln fallen die Anwendungsfälle *Artikel anlegen*, *Artikel Übersicht*, *Artikel Suchen und Filtern*, *Artikel bearbeiten*, *Artikel schließen* und *Mediathek*. Das *Datenmodell* für einen konkreten Artikel (auf dem die nachfolgenden Anwendungsfälle basieren) ist wie folgt aufgebaut:

id	Eindeutige ID eines konkreten Artikels. Autoincrement. Primärschlüssel.
author_id	ID des Autors, der diesen Artikel/die Revision dieses Artikels erstellt hat.
date	Timestamp, wann dieser Artikel/die Revision dieses Artikels erstellt wurde.
content	Frei eingebbarer Inhalt des Artikels in Form von XHTML.
title	Frei eingebbarer Titel dieses Artikels.
name	Eindeutiger Name dieses Artikels in URL-konformer Schreibweise, generiert aus dem Titel und evtl. weiteren Merkmalen.
status	Ein Artikel kann folgende Status haben: • „Neu“ • „In Bearbeitung“ • „Geschlossen“ • „Freigegeben“
comment_status	Status zeigt an, ob Kommentare abgegeben werden dürfen. Dieser kann „yes“ oder „no“ sein.
type	Typ des Artikels. Dieser kann „article“ oder „revision“ sein.
revision_id	Verweis auf die ID des Artikels, zudem dieser Artikel eine Revision bildet. Für die erste Version eines Artikels ist dieses Datenbankfeld leer.
order	Reihenfolge, in der dieser Artikel erscheinen soll, sobald sich mehrere Artikel auf einer Ebene (Node) befinden.
indicator_id	Verweis auf die Indikatoren, die in diesem Artikel beschrieben werden (Verweis auf die Tabelle, die die n:m-Beziehung der Indikatoren zu den Artikeln abbildet.
tnode_id	Verweis auf den zugehörigen Node aus dem Schema.

Tabelle 9.36.: Datenbanktabelle „Article“

Sobald ein NB erstellt wurde, kann ein Benutzer dem NB neue Artikel hinzufügen. Ein neuer Artikel muss einem Kapitel (Node) zugeordnet werden, die weiteren Bearbeitungsschritte werden durch den folgenden Anwendungsfall abgedeckt. Der Anwendungsfall *Artikel bearbeiten* beschreibt, in welchem Umfang ein Artikel editiert welchen kann. Zunächst muss ein neuer oder ein bestehender Artikel zum Bearbeiten ausgewählt werden. Die Bearbeitungsoberfläche wird in Abbildung 9.8 beispielhaft dargestellt. Nun können verschiedene Daten für einen Artikel bearbeitet werden:

- Titel
- Beschreibung
- Tags
- Links (z.B. für weitergehende Informationen)
- Inhalt
- Status
- Indikatoren

Der Inhalt kann als Freitext in das Content-Feld eingegeben oder als Verweis eingefügt werden (z.B. Bilder/Videos, siehe Mediathek 9.2.2). Der dazu genutzte Editor ist der TinyMCE-Editor [Tin10]. Dieser Editor kann einfach in Grails integriert werden und unterstützt die benötigten Formatierungen. Unterstützte Formatierungen des Editors sind:

- Schriftart ändern
- Fett/kursiv/unterstrichen/durchgestrichen/
- Schriftfarbe ändern
- Schrifthintergrundfarbe ändern
- Auflistung mit Bullet Points/ Nummerierung
- Einfügen von multimedialen Inhalten

Weiterhin können für einen Artikel Indikatoren und Inhaltsrichtlinien bearbeitet werden, über die im Artikel berichtet werden kann. Indikatoren werden dem Artikel bei Erstellung eines NBs auf einem bestimmten Schema automatisch zugewiesen. Diese können obligat oder optional sein. Zusätzlich kann die Liste der Indikatoren um eigene Indikatoren erweitert werden. Sie können über eine Liste ausgewählt und dem Artikel zugeordnet werden. Die Angabe ist verpflichtend, damit der Artikel (und damit der gesamte Bericht) gegen das Schema validiert werden kann, um zu prüfen, ob über alle notwendigen Indikatoren berichtet wurde. Für Indikatoren kann im Text eines Artikels eine Referenz eingefügt werden, über die der Wert automatisch eingefügt wird. Der Wert wird an einer zentralen Stelle bearbeitet.
Sobald ein Artikel durch einen Benutzer bearbeitet wird, wird dieser für die Bearbeitung durch andere Benutzer gesperrt, dabei wird eine Sperrzeit von 120 Minuten verwendet. Sobald der Benutzer eine Aktion durchführt bzw. Text schreibt, wird die Sperrzeit verlängert. Die Zeit wird dazu in Timestamps in der Tabelle Article-Meta gespeichert.
Wenn andere Benutzer währenddessen den Artikel bearbeiten wollen, erhalten diese eine

Meldung, dass der Artikel nicht bearbeitet werden kann, mit dem Hinweis, durch wen der Artikel gesperrt wurde.
Der bearbeitete Artikel kann in einer Vorschau dargestellt werden, dies geschieht direkt im Frontend. Die *Artikel-Vorschau* zeigt alle Informationen zu dem bearbeiteten Artikel: Titel, Content und Indikatoren. Sobald ein Artikel nach vorgenommenen Änderungen gespeichert wird, wird eine Revision des alten Zustands gespeichert (der Artikel wird versioniert). Dazu wird der alte Zustand in eine neue Tabellenzeile gespeichert und der Vermerk auf „Revision“ gesetzt. Zusätzlich wird in der Spalte „Revision_id“ ein Verweis auf den ursprünglichen Artikel gespeichert. Der neue Artikel wird in die Tabellenzeile des ursprünglichen Artikels gespeichert. Dabei werden neben dem Inhalt weitere bearbeitete Datenbankfelder aktualisiert (z.B. Autor, Datum etc.). Durch diese Vorgehensweise ändert sich die ID eines Artikels nicht.
Folgend die Anwendungsfallbeschreibung für *Artikel speichern*:

Name	Artikel speichern
Beschreibung	Alle vorgenommene Änderungen an einem Artikel sollen gespeichert speichern. Dabei soll die Revision des Artikels beachtet werden, d.h. neue Tabellenzeile mit „Version“, Verweis auf eigentlichen Artikel geben, Angaben zu Autor, Datum und Uhrzeit.
Beteiligte Akteure	Client, Server, ArticleController, Datenbank
Auslöser	Benutzer klickt auf „Artikel speichern“.
Vorbedingungen	Benutzer ist eingeloggt und verfügt über die entsprechende Rechte. Ein Artikel wurde angelegt oder ein bereits existierender Artikel wurde geöffnet.
Nachbedingung/Ergebnis	Der Artikel wurde in der Datenbank vollständig gespeichert.
Standardablauf	siehe Abbildung 9.9

Tabelle 9.37.: Anwendungsfall „Artikel speichern“

Der Benutzer kann einen Artikel *Löschen*, dabei wird ein Artikel nicht wirklich gelöscht, sondern nur dessen Status auf *Geschlossen* gesetzt. Ein geschlossener Artikel ist im NB nicht sichtbar. Ein geschlossener Artikel kann bei Bedarf für die Bearbeitung bzw. Änderung nachträglich wieder geöffnet werden, dazu wird der Status auf „in Bearbeitung“ gesetzt.

Indikatoren verwalten

Indikatoren werden bestimmten Artikeln zugeordnet um Aussagen über bestimmte Zustände zu treffen (z.B. Wasserverbrauch in Litern im Jahr 2009). Aus der Liste der NBs kann zu jedem NB eine Liste von zugehörigen Indikatoren angezeigt werden. Auch aus der Liste der Artikel eines NBs kann wiederum zu jedem Artikel eine Liste von Indikatoren für diesen Artikel angezeigt werden. Beim Erstellen eines neuen NBs werden die Indikatoren des dazu ausgewählten Schemas den jeweiligen Nodes (hier: Artikeln) zugeordnet. Diese Indikatoren müssen mit einem Wert gefüllt werden, damit der jeweilige NB gegen sein jeweiliges Schema valide ist. Beim Bearbeiten eines Artikels können darüber hinaus eigene Indikatoren erstelllt und diesem Artikel zugeordnet werden. Beim Bearbeiten können den einzelnen Indikatoren auch Werte zugewiesen werden. Die Werte einzelner Indikatoren

sind frei eingebbar. Für einzelne Indikatoren können spezifische Vorschläge für Werte hinterlegt werden. Diese Vorschläge stammen aus externen Systemen. Diese können entweder SAP-Systeme (BAPI-Anbindung) oder ODBC-Datenquellen (z.B: StudIp) sein. Sobald die Werte aller Pflichtindikatoren gesetzt wurden, ist der NB valide gegenüber seinem Schema.

Mediathek

Die *Mediathek* erlaubt die Verwaltung von Bildern und multimedialen Dateien wie Grafiken und Videos, die mit einem Titel, einer Beschreibung und einem ALT-Text versehen werden können. Weiterhin werden das Datum der Erstellung, die Dateiendung und der Typ der Datei automatisch gespeichert. Der Typ ist wichtig, um die Dateien im Artikeleditor entweder Bildern oder anderen Multimediadateien zuordnen zu können.
Multimediale Daten werden ausschließlich in der Mediathek gespeichert und verwaltet, um sie in einem Artikel referenzieren zu können. Die Speicherung geschieht nicht über die Datenbank, sondern über ein Dateisystem, dies ist nötig, um sie aus dem Rich Text Editor heraus zu refrenzieren. Die Dateien werden in einem Ordner gespeichert, der den gleichen Namen hat wie der Benutzer, der die Datei hochgeladen hat und ein Unterordner des Ordners „Mediathek“ ist. Die Dateinamen werden von evtl. vorhandenen Sonderzeichen bereinigt, diese werden durch „_“ ersetzt. Somit wird z.B. eine Datei mit dem Namen „Bild.jpg“, die vom User „Nutzer“ hochgeladen wurde, im Verzeichnis „/Mediathek/Nutzer/Bild.jpg“ gespeichert. Bei Bildern wird zusätzlich ein verkleinertes Vorschaubild gespeichert, im Unterordner „imgthumbs“, zu der Datei „Bild.jpg“ wird das Vorschaubild „/Mediathek/Nutzer/imgthumbs/thumbBild.jpg“ gespeichert.
Die genutzten Dateien können vom lokalen Dateisystem auf den Server übertragen werden, als auch von Quellen aus dem Internet kopiert werden. Bei der Kopie der Datei wird zusätzlich die URL gespeichert, um einen Überblick über die Herkunft externer Daten zu behalten, beide Verfahren sind über die Weboberfläche von STORM möglich.
Die Organisation der Multimediadateien bietet die Möglichkeit, diese nach Name, Dateiendung, Beschreibung und Besitzer zu sortieren, in den Attributen der Dateien zu suchen, sowie die Dateien einem speziellen Ansichten-System zuzuordnen. Dieses Ansichten-System bildet die Strukturen aller vorhandenen NBs als Übersichtsliste der Artikel ab. Jede einzelne Mediendatei kann einem oder mehreren dieser Artikel hinzugefügt werden, sie wird dann in der Übersicht bei diesem Artikel angezeigt. Bei Bedarf kann sie auch wieder aus der Übersicht entfernt werden. Der Zweck dieser Übersicht ist rein informativ, die Mediendatei selbst wird dabei nicht verändert.

Es folgen die Anwendungsfälle für die Mediathek:

Name	Datei hochladen
Beschreibung	Eine Datei wird vom User von seinem lokalen Dateisystem auf den Server, auf dem die Anwendung läuft, hochgeladen.
Beteiligte Akteure	User, Client, Server, Media-Controller
Auslöser	User ruft „Medien anlegen“ auf, füllt das Formular aus und klickt auf „Anlegen“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte.

Nachbedingung/Ergebnis	Die Datei ist auf dem Server gespeichert unter dem Pfad „/Mediathek/Username/Dateiname“. Falls es eine Bilddatei ist, ist zusätzlich ein Vorschaubild unter „/Mediathek/Nutzername/imgthumbs/thumbDateiname“ gespeichert.
Standardablauf	Die Datei wird auf den Server geladen und im entsprechenden Verzeichnis abgespeichert (Controller Media, Action save). Sonderzeichen werden durch „_“ ersetzt.

Tabelle 9.38.: Anwendungsfall „Datei hochladen“

Name	Datei runterladen
Beschreibung	Eine Datei wird vom User von einer Internetquelle auf den Server, auf dem die Anwendung läuft, hochgeladen.
Beteiligte Akteure	User, Client, Server, Media-Controller
Auslöser	User ruft „Medien anlegen“ auf, füllt das Formular aus und klickt auf „Anlegen“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte.
Nachbedingung/Ergebnis	Die Datei ist auf dem Server gespeichert unter dem Pfad „/Mediathek/Username/Dateiname“. Falls es eine Bilddatei ist, ist zusätzlich ein Vorschaubild unter „/Mediathek/Nutzername/imgthumbs/thumbDateiname“ gespeichert.
Standardablauf	Die Datei wird auf den Server geladen und im entsprechenden Verzeichnis abgespeichert (Controller Media, Action download). Sonderzeichen werden durch „_“ ersetzt.

Tabelle 9.39.: Anwendungsfall „Datei runterladen“

Name	Datei bearbeiten
Beschreibung	Der User bearbeitet die Attribute der Datei.
Beteiligte Akteure	User, Client, Server, Media-Controller
Auslöser	User ruft über „Medien verwalten“ die Datei auf, bearbeitet das Formular und klickt auf „Speichern“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte.
Nachbedingung/Ergebnis	Die neuen Attribute der Datei sind gespeichert.
Standardablauf	Über den Controller Media, Action edit werden die Attribute neu gesetzt.

Tabelle 9.40.: Anwendungsfall „Datei bearbeiten“

Name	Datei löschen
Beschreibung	Der User löscht die Datei vom Server.

Beteiligte Akteure	User, Client, Server, Media-Controller
Auslöser	User ruft über „Medien verwalten“ die Datei auf und klickt auf das Löschen-Icon oder über das Bearbeiten der Datei auf „Löschen“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte. Die Datei darf nicht referenziert sein.
Nachbedingung/Ergebnis	Die Datei ist vom Server gelöscht. Bei Bildern auch das Vorschaubild.
Standardablauf	Über den Controller Media, Action delete wird die Datei gelöscht.

Tabelle 9.41.: Anwendungsfall „Datei löschen“

Name	Datei in Artikel einbinden
Beschreibung	Der User benutzt eine Datei aus der Mediathek im Artikeleditor.
Beteiligte Akteure	User, Client, Server, Rich Text Editor, Article-Controller
Auslöser	User benutzt den Rich Text Editor, um eine Datei einzubinden.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte. Datei ist auf dem Server.
Nachbedingung/Ergebnis	Die Datei ist im bearbeiteten Artikel referenziert.
Standardablauf	Der Rich Text Editor fügt die Referenz auf die Datei im Artikel ein. Beim Speichern des Artikels werden die benutzten Dateien markiert und es wird in den Attributen der Datei eingetragen, wo sie benutzt wurde.

Tabelle 9.42.: Anwendungsfall „Datei in Artikel einbinden“

Name	Datei in Ansichten-System einbinden
Beschreibung	Der User ordnet eine Datei einem Artikel im Ansichten-System zu.
Beteiligte Akteure	User, Client, Server, Media-Controller
Auslöser	User klickt bei der Datei auf „Zu Sicht hinzufügen“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechende Rechten. Datei ist auf dem Server.
Nachbedingung/Ergebnis	Die Datei ist in der Ansicht des Artikels referenziert.
Standardablauf	Der Media-Controller, Action addToView, fügt die Datei der Ansicht hinzu.

Tabelle 9.43.: Anwendungsfall „Datei in Ansichten-System einbinden“

Name	Datei aus Ansichten-System entfernen
Beschreibung	Der User entfernt eine Datei aus dem Ansichten-System.
Beteiligte Akteure	User, Client, Server, Media-Controller

Auslöser	User klickt bei der Datei auf „aus Sicht entfernen“.
Vorbedingungen	User ist eingeloggt und verfügt über die entsprechenden Rechte. Die Datei ist in der Ansicht des Artikels referenziert.
Nachbedingung/Ergebnis	Die Datei ist nicht mehr in der Ansicht des Artikels referenziert.
Standardablauf	Der Media-Controller, Action removeFromView, entfernt die Datei aus der Ansicht.

Tabelle 9.44.: Anwendungsfall „Datei aus Ansichten-System entfernen“

Externe Systeme

Um Vorschläge für Werte für Indikatoren zu bekommen, können externe Systeme angebunden werden. Es sind Verbindungen zu SAP-Systemen sowie zu ODBC-Datenbanken möglich. Dadurch können bestimmte Daten wie z.B. die aktuelle Anzahl der Professoren ausgelesen werden. Das Ergebnis der Anfrage wird dem Benutzer, welcher die entsprechenden Rechte hat, Indikatoren zu pflegen, als Vorschlag präsentiert. Dieser kann nun entscheiden, ob der vom externen System vorgeschlagene Wert übernommen werden soll oder ein anderer Wert für den Indikator gesetzt werden soll. Die Pflege der Indikatoren-Werte erfolgt im Eingabeeditor beim Bearbeiten der Indikatoren eines bestimmten Artikels. Die Pflege der externen Systeme erfolgt im Administrationsbereich. Für SAP Systeme müssen die Verbindungsparameter angegeben werden (Client, Login, Passwort, Locale, URL, System). Zusätzlich können Name und Beschreibung eingegeben werden. Es können beliebig viele unterschiedliche SAP-Systeme eingebunden werden. Für jedes SAP-System können wiederum beliebig viele BAPIs angesprochen werden. Diese werden immer einer bestimmten SAP-Verbindung zugeordnet und enthalten als Parameter den Namen der BAPI die berechnet werden soll sowie eine Liste von Eingabe-Parametern, die Export-Struktur sowie den Export-Wert. Zusätzlich können ein Name und eine Beschreibung vergeben werden.

FÜR ODBC-Datenquellen müssen die URL, der Treiber (z.B. com.mysql.jdbc.Driver), der User und das Passwort angegeben werden. Zusätzlich können Name und Beschreibung vergeben werden. Auf Basis der Verbindungsdaten kann eine Verbindung zu dem jeweiligen ODBC-System hergestellt werden. Es können beliebig viele unterschiedliche Datenbankverbindungen angegebenen werden. Für jede Datenbankverbindung können wiederum beliebig viele SQL-Statements definiert werden. Diese sind immer einer Datenbankverbindung zugeordnet und enthalten das eigentliche SQL-Statement. Zusätzlich können Name und Beschreibung angegeben werden.

Jedem Indikator kann bei Erstellung eine BAPI und ein SQl-Statement zugeordnet werden. Beim Bearbeiten des Indikator-Wertes aus dem Eingabeeditor heraus werden die Werte von BAPI und SQL-Statement berechnet und dem Benutzer als Vorschlagswert präsentiert.

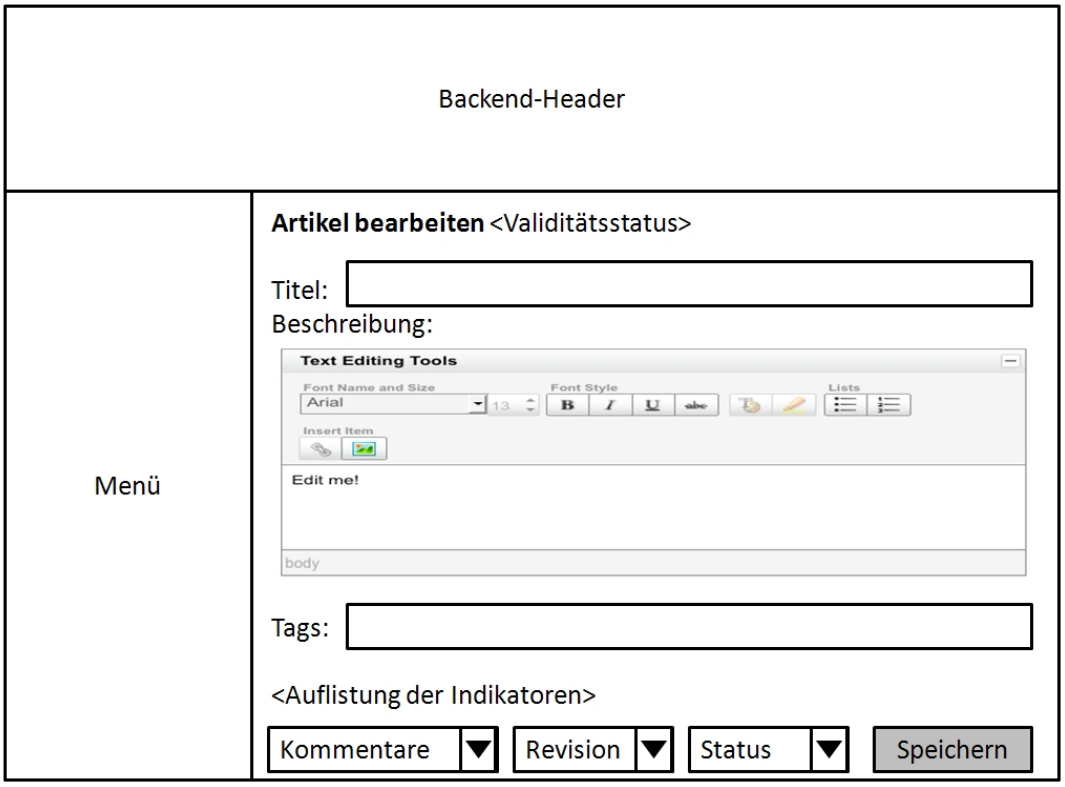

Abbildung 9.8.: Oberflächenentwurf „Artikel bearbeiten“

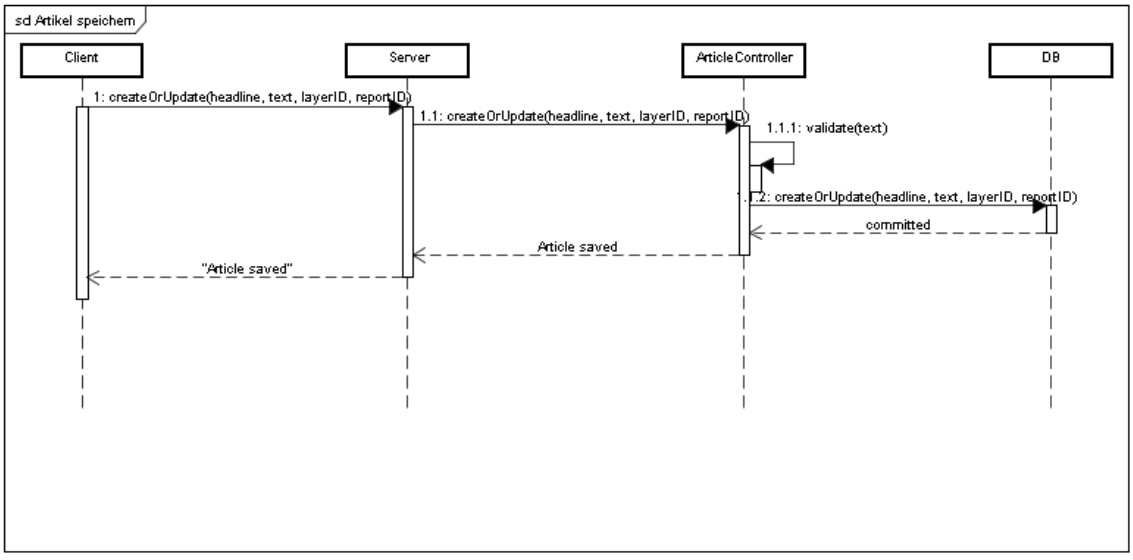

Abbildung 9.9.: Sequenzdiagramm „Artikel speichern“

9.3. Öffentlicher Bereich

Dieses Kapitel beschreibt die Möglichkeiten eines normalen Benutzers, Nachhaltigkeitsberichte zu lesen, zu konvertieren und zu speichern, sowie diese zu bewerten und zu kommentieren. Zusätzlich wird die Unterstützung des Benutzers mit einem Tag-Cloud und einer Recommender Engine beschrieben.

9.3.1. Darstellung der Berichte/Artikel

Sobald die Startseite aufgerufen wird, sieht der Benutzer in der Seitenleiste, an oberste Stelle, den zuletzt herausgegebene Bericht. In diesem Bereich hat der Benutzer die Möglichkeit die einzelnen Artikel des Berichtes oder aber auch andere Berichte und deren zugehörige Artikel einzusehen. Nachdem der Anwender sich für einen Bericht entschieden hat, kann er diesen durch einen Klick auf den Bericht öffnen. Die Artikelstruktur des Nachhaltigkeitsberichtes wird angezeigt und der Benutzer kann dann innerhalb dieser Artikelstruktur zu bestimmten Artikeln wechseln. Standardmäßig werden bereits die Kapitel des neuesten Nachhaltigkeitsberichtes angezeigt. Nach Auswahl eines älteren NB werden diese Kaptiel jedoch ausgeblendet und die Kapitel des dann gewählten Berichtes eingeblendet. Aus dem gewählten Bericht kann der Anwender nun zielgerichtet Artikeln auswählen und diese zu einem eigenen, individuell zusammengestellten Infokorb zusammenfassen. Damit kann der Benutzer gleichzeitig einen neuen, personalisierten Bericht erstellen, der an den eigenen Informationsbedürfnisse zugeschnittenen ist und in Form einer PDF-Datei heruntergeladen oder in HTML Format auf eine Seite präsentiert werden kann. Die jeweilige Anwendungsfallbeschreibung hierzu kann nachfolgenden Zusammenstellungen entnommen werden:

Name	Berichte anzeigen
Beschreibung	Nachdem der Benutzer die Anwendung betreten kann werden ihm alle verfügbaren Bericht angezeigt
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer startet Storm
Vorbedingungen	Es muss in Datenbank mindestens ein Bericht existieren
Nachbedingung/Ergebnis	Der Benutzer sieht die aktuellen Berichte
Standardablauf	Das System zeigt den aktuellsten Bericht an.

Tabelle 9.45.: Anwendungsfall: „Berichte anzeigen“

Name	Bericht auswählen
Beschreibung	Der Benutzer hat die Möglichkeit, einen Nachhaltigkeitsbericht innerhalb der Website auszuwählen und die zugehörigen Artikel anzuzeigen.
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer möchte einen Nachhaltigkeitsbericht lesen
Vorbedingungen	Es muss in Datenbank mindestens ein Bericht existieren
Nachbedingung/Ergebnis	Der Benutzer kann den Bericht auswählen.

Standardablauf	Das System erhält die Benutzerwahl und zeigt ausgewählten Bericht an.

Tabelle 9.46.: Anwendungsfall: „Bericht auswählen“

Name	Artikel auswählen
Beschreibung	Der Benutzer hat die Möglichkeit, einzelne Artikel des jeweiligen Nachhaltigkeitsberichtes anzusehen.
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer wählt Kapitel eines Nachhaltigkeitsberichtes.
Vorbedingungen	Es muss zuvor ein NB und ein Kapitel ausgewählt worden sein und in Datenbank muss mindestens ein Artikel existieren.
Nachbedingung/Ergebnis	Der Benutzer kann den Artikel lesen.
Standardablauf	Das System erhält die Benutzerwahl und zeigt alle Artikel eines gewählten Kapitels des gewählten NB in Form einer HTML Seite an.

Tabelle 9.47.: Anwendungsfall: „Artikel auswählen“

9.3.2. Infokorb

Der Infokorb ist auf dem Prinzip eines Einkaufskorbes konzipiert und dient dazu vom Benutzer ausgewählte Artikel aufzunehmen. Der zusammengestellte personalisierte Infokorb kann als PDF, XML oder HTML anzeigt, exportiert und gespeichert werden.

Der Infokorb wird folgendermaßen benutzt:

In der Artikelansicht kann der Benutzer mit einem Klick auf das entsprechende Symbol den aktuellen Artikel seinem Infokorb hinzufügen. Der Infokorb kann mit beliebig vielen Artikeln befüllt werden. Im Bereich des Infokorbs kann der Benutzer auswählen zwischen der Anzeige des aktuellen Infokorbs, einer Auflistung von Infokörben, die durch andere Benutzer zusammen- und zur Verfügung gestellt wurden und dem Erstellen eines neuen Infokorbs. Mit einer Anmeldung können Personalisierungsaspekte von STROM in Form eines „Meine Infokörbe“-Bereichs genutzt werden.

Unter dem Menüpunkt „Infokorb anzeigen“ werden alle Artikel, die der Benutzer ausgewählt hat aufgelistet. Diese Liste ist anfangs leer. Sofern alle Artikel dieses Infokorbs aus dem selben Nachhaltigkeitsbericht stammen, werden sie nach deren Datenbank ID im Infokorb hinterlegt, die ID-s entwprechen der vom Schema vorgegebenen Ordnung. Wenn der Infokorb Artikel aus mehreren Nachhaltigkeitsberichten enthält, werden die Artikel in der Reihenfolge angezeigt, in der sie vom Benutzer in den Infokorb eingefügt wurden.

Name	Infokorb anzeigen
Beschreibung	Der Benutzer hat die Möglichkeit den Infokorb anzuzeigen.
Beteiligte Akteure	Benutzer, System
Auslöser	Der Benutzer möchte Infokorb anzeigen und klickt dazu auf den entsprechenden Link

Vorbedingungen	-
Nachbedingung/Ergebnis	Infokorb wird angezeigt
Standardablauf	Das System empfängt die Benutzerinteraktion zur Anzeige der Infokörbe und gibt eine Liste von aktuellen Inforkörben aus.

Tabelle 9.48.: Anwendungsfall: „Infokorb anzeigen“

Name	Artikel zum Infokorb hinzufügen
Beschreibung	Der Benutzer hat die Möglichkeit, Artikel zum Infokorb hinzuzufügen
Beteiligte Akteure	Benutzer, System
Auslöser	Der Benutzer möchte einen Artikel in den Infokorb hinzufügen und klickt dazu beim entsprechenden Artikel auf den entsprechenden Link
Vorbedingungen	Es muss mindestens ein Artikel existieren
Nachbedingung/Ergebnis	Ein Artikel wird zum Infokorb hinzugefügt
Standardablauf	siehe Abbildung 9.10

Tabelle 9.49.: Anwendungsfall: „Artikel zum Infokorb hinzufügen“

STORM als offene Plattform bietet die Möglichkeit der anonymen Benutzung. Ohne Anmeldung kann ein Infokorb zusammengestellt, in Form von HTML, XML oder PDF betrachtet und nach Wunsch heruntergeladen werden.

Name	PDF/XML aus Artikel erzeugen
Beschreibung	Der Benutzer hat die Möglichkeit, Artikel als PDF oder XML Dokument zu erzeugen.
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer möchte PDF/XML aus dem Artikel erzeugen. Dazu wählt er die entsprechende Funktion per eingebauten Button
Vorbedingungen	Es muss sich ein Artikel angezeigt werden.
Nachbedingung/Ergebnis	Der Benutzer kann die PDF/XML-Ausgabe betrachten.
Standardablauf	Anwendung holt Artikel aus DB, transformiert diese per PDF oder XML Plugin und zeigt sie auf der Webseite an.

Tabelle 9.50.: Anwendungsfall: „PDF/XML aus Artikel erzeugen“

Dem gegenüber hat der angemeldete Benutzer zusätzliche Personalisierungsoptionen in seinem Infokorb. Im Gegensatzt zur anonymen Benutzung, wird bei eine Neuerstellung eines Infokorbs der Vorherige nicht automatisch übeschrieben. Individuell zusammengestellte Infokörbe können gespeichert werden, um diese bei Bedarf zu nutzen, oder komplett zu entfernen. Zusätzlich können Infokörbe modifiziert werden. Angefangen beim Namen des Infokorbs bis hin zum Inhalt kann jederzeit flexibel eine Veränderung vorgenommen werden. Desweiteren kann der Infokorb, für andere Benutzer sichtbar gemacht oder redak-

tionell in der Sidebar veröffentlicht werden.

Technischer Hintergrund:

Sobald ein Artikel in den Infokorb verschoben wird, schickt die View diese Information an den Controller, dieser gibt es weiter an das Model. Das Model wiederum entscheidet, was passiert. In diesem Fall wird gespeichert, welche Artikel der registrierte Benutzer in den Infokorb verschoben hat, da die Zuordnung Artikel zu Benutzer durch die Recommender Engine ausgewertet werden kann. Bei anonymen Benutzern wird der Inhalt des Infokorbs nur dann gespeichert, wenn mehr als ein Artikel darin ist, in diesem Fall wird die Verknüpfung der Artikel gespeichert.

Bei weiteren Veränderungen (sowohl für registrierte als auch anonyme Benutzer) werden diese entsprechend gespeichert. Für die Infokörbe der Benutzer werden folgende Informationen gespeichert:

Infokorb-Informationen:

- Infokorb-ID
- Benutzer-ID
- Name (wird gesetzt, wenn Benutzer Infokorb speichert)
- Timestamp Infokorb gespeichert (wird gesetzt, wenn Benutzer Infokorb speichert)
- Infokorb aktiv (wird auf `false` gesetzt, wenn Benutzer gespeicherten Infokorb löscht. Notwendig, um auch gelöschte Infokörbe noch für Recommender Engine nutzen zu können)

Infokorb-Inhalte:

- Infokorb-ID
- Artikel-ID
- Timestamp Artikel zu Infokorb hinzugefügt

9.3.3. Kommentarfunktion

Artikel von Nachhaltigkeitsberichten können von registrierten Benutzern kommentiert werden. Um dieses zu tun, klickt der Benutzer auf „Kommentar hinzufügen“. In einem interaktiven Textbereich kann nun ein Kommentar eingegeben werden, der beim Absenden dem Artikel hinzugefügt wird. Jeder Benutzer kann beliebig viele Kommentare abgeben und auch andere Kommentare kommentieren. Aus technischer Sicht ist dies ein Insert in die Datenbank, vobei folgende Attribute gespeichert werden:

- Artikel-ID
- Benutzer-ID
- Kommentar-ID
- Timestamp
- Kommentartext

Name	Artikel kommentieren
Beschreibung	Der Benutzer kommentiert einen Artikel oder Kommentar.

Beteiligte Akteure	User, Client, Server, Datenbank.
Auslöser	Benutzer klickt auf "ŽKommentar speichern“.
Vorbedingungen	Anwender hat einen Artikel aufgerufen und einen Kommentar hierzu eingetragen.
Invarianten	Integrität des Artikels: weder der Inhalt noch die Darstellung darf beeinflusst werden.
Nachbedingung/Ergebnis	Der Kommentar für den Artikel wurde gespeichert.
Standardablauf	Die Benutzerinteraktion zur Kommentierung eines Artikels bzw. des Kommentars wird vom Controller empfangen und Inhalt, sowie Namen des Kommentars in der Datenbank gespeichert. Anschließend wird der Kommentar in der View angezeigt.

Tabelle 9.51.: Anwendungsfall „Artikel kommentieren“

9.3.4. Bewertungsfunktion

Artikel und Nachhaltigkeitsberichte können bewertet werden, die Bewertung erfolgt über anklickbare Sterne, 1 Stern entspricht schlecht, 5 Sterne entsprechen sehr gut. Sowohl registrierte als auch anonyme Benutzer können bewerten. Um eine Manipulation durch anonyme Bewertungen zu verhindern, fallen alle anonymen Bewertungen zusammen als eine einzige zu Gewicht. Pro Artikel kann eine Bewertung abgegeben, jedoch zu einem späteren Zeitpunkt auch verändert werden. Die alte Bewertung des Benutzers wird dabei überschrieben.
Für die Bewertung werden folgende Informationen gespeichert:

- Artikel-ID
- Benutzer-ID
- Bewertung (Anzahl Sterne)

Bei der ersten Bewertung (und bei anonymen) ist dies ein Insert, sonst ein Update auf der Datenbank.

Name Beschreibung	Artikel bewerten Der Benutzer bewertet den Artikel.
Beteiligte Akteure	User, Client, Server, Datenbank.
Auslöser	Benutzer klickt auf "ŽArtikel bewerten“.
Vorbedingungen	keine
Invarianten	Integrität des Artikels: weder der Inhalt noch die Darstellung darf beeinflusst werden.
Nachbedingung/Ergebnis	Die Bewertung für den Artikel wurde gezählt.
Standardablauf	Der Benutzer klickt auf einer Bewertungsskala seine Bewertung für den Artikel an.
Alternative Ablaufschritte	Ist die Bewertung des Artikels nicht möglich, wird die Aktion abgebrochen und dem Benutzer der Mißerfolg mitgeteilt.

Tabelle 9.52.: Anwendungsfall „Artikel bewerten“

9.3.5. Tags vergeben

Ein registrierter Benutzer kann einem Artikel ein Stichwort (Tag) zuordnen, das beschreibend für den Artikel steht. Anhand dieser Tags werden die Tag-Cloud und die Recommender Engine beeinflusst.
Um einen Tag hinzuzufügen klickt ein Benutzer auf „Tag hinzufügen“. In einem einfachen Textfeld kann nun ein Tag eingegeben werden, der beim Absenden dem Artikel hinzugefügt wird. Jeder angemeldete Benutzer kann beliebig viele Tags hinzufügen, der anonymen Benutzer kann keine Tags vergeben, jedoch vorhandene sehen und nutzen.

Name Beschreibung	Artikel taggen Der Benutzer ordnet dem Artikel ein Stichwort zu.
Beteiligte Akteure	User, Client, Server, Datenbank.
Auslöser	benutzer klickt auf "Tag hinzufügen“.
Vorbedingungen	Der Benutzer muss angemeldet sein.
Invarianten	Integrität des Artikels: weder der Inhalt noch die Darstellung darf beeinflusst werden.
Nachbedingung/Ergebnis	Der Artikel hat einen Tag mehr als vorher.
Standardablauf	Der Benutzer gibt in ein dafür vorgesehenes Textfeld ein Stichwort ein, dass beim Absenden dann den Tags hinzugefügt wird.
Alternative Ablaufschritte	Ist die Speicherung des Tags nicht möglich, wird die Aktion abgebrochen und dem Benutzer der Mißerfolg mitgeteilt.

Tabelle 9.53.: Anwendungsfall „Tags für Artikel vergeben“

9.3.6. PDF-Generator

Die Visualisierung der ausgewählten Artikel kann in Form von Portable Document Format (PDF) erfolgen. Dadurch wird eine Unabhängigkeit von Plattformen, Betriebssystemen und Anwendungsprogrammen mit einer originalgetreuen Visualisierung der elektronischen Dokumente geschaffen.

Die Generierung einer PDF Datei wird durch ein Grails Plugin realisiert. Mit Hilfe dieses Plugins wird es dem Benutzer der Software ermöglicht, ausgewählte Inhalte der Nachhaltigkeitsberichte zu einem PDF Dokument zusammenzufassen. Der Benutzer muss dazu nur auf die Schaltfläche zur Generierung klicken und evtl. noch die von ihm gewünschte Struktur anpassen.

Name	PDF aus Infokorb erzeugen
Beschreibung	Der Benutzer hat die Möglichkeit, Artikel, der er in seinen Infokorb gelegt hat als PDF Dokument zu erzeugen
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer möchte PDF aus dem Infokorb erzeugen. Dazu wählt er die entsprechende Funktion per eingebauten Button

Vorbedingungen	Es muss sich mindestens ein Artikel im Infokorb befinden
Nachbedingung/Ergebnis	Der Benutzer kann die PDF-Ausgabe betrachten
Standardablauf	
Alternative Ablaufschritte	-
Hinweise	Da die Abläufe für die PDF-Generierung gleich ablaufen, wurden diese in diesem einen Anwendungsfall zusammen beschrieben, dabei handelt es sich jedoch immer um getrennte Aktionen, d.h. es wird immer entweder ein PDF-Dokument erzeugt

Tabelle 9.54.: Anwendungsfall: „PDF aus Infokorb erzeugen“

9.3.7. XML

Die Extensible Markup Language (XML), zielt hierarchische Darstellung strukturierter Datensätze an. Grails unterstützt die Modifizierung (Erstellung und Änderung) von XML Dateien. XML wird innerhalb des Projekts zum Extrahieren von Daten aus dem System, zum Transfer von Daten zwischen Webservern und für den Transfer zwischen Webserver und Client genutzt.

Grails bietet durch den `StreamingMarkupBiulder` eine komfortable möglich, XML aus Domain-Klassen zu generieren. Storm bietet zu den Bereichen Artikel, Schema und Infokorb die Möglichkeit, Informationen als XML abzufragen. Ein XML kann von einem Client über eine entsprechende URL und einem daraus resultierenden Action-Aufruf angefordert werden. Dabei wird die Action xml des jeweiligen Controllers (ArticleController, TReportController, ReportController oder TNodeController) aufgerufen. Die Action `xml` erwartet die `ID` des Objekts, welches als XML exportiert werden soll. Die allgemeine Syntax für XML-Anfragen ist `...Storm/<Controller>/xml/<ID>`. Es folgt nun eine detaillierte Darstellung konkreter Funktionen, um ein XML zu generieren.

Artikel als XML

Artikel können über die URL Storm/article/xml/<ID> als XML angefragt werden.

Schema als XML

Ein Schema kann über die URL Storm/TReport/xml/<ID> als XML angefragt werden. Einem Schema-XML liegt dabei das folgende XML-Schema zugrunde:

Einzelne Ebenen des Schemas können über die URL Storm/TNode/xml/<ID> als XML angefragt werden. Indikatoren einer Ebene können über die URL Storm/indicator/xml/<ID> als XML angefragt werden.

Nachhaltigkeitsbericht als XML

Nachhaltigkeitsberichte können über die URL Storm/report/xml/<ID> als XML angefragt werden.

Name	XML aus Infokorb erzeugen

Beschreibung	Der Benutzer hat die Möglichkeit, Artikel, der er in seinen Infokorb gelegt hat als XML Dokument zu erzeugen
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer möchte XML aus dem Infokorb erzeugen. Dazu wählt er die entsprechende Funktion per eingebauten Button
Vorbedingungen	Es muss sich mindestens ein Artikel im Infokorb befinden
Nachbedingung/Ergebnis	Der Benutzer kann die XML-Ausgabe betrachten
Standardablauf	
Alternative Ablaufschritte	-
Hinweise	Da die Abläufe für die XML-Generierung gleich ablaufen, wurden diese in diesem einen Anwendungsfall zusammen beschrieben, dabei handelt es sich jedoch immer um getrennte Aktionen, d.h. es wird immer entweder ein PDF-Dokument erzeugt

Tabelle 9.55.: Anwendungsfall: „XML aus Infokorb erzeugen“

9.3.8. RSS-Schnittstelle

Es wird die Möglichkeit geben, RSS-Feeds zu abonnieren. Es wird RSS-Feeds geben zu Nachhaltigkeitsberichten als Ganzes, zu einzelnen Artikeln, den Kommentaren und zu den News.
Der Benutzer kann bestimmen, ob er in den Feeds direkt den Inhalt lesen kann, oder ob er nur darüber informiert wird, dass eine Änderung vorliegt.
Für das System der RSS-Feeds wird ein Plugin für Grails benutzt.

Name	RSS-Feed abonnieren
Beschreibung	Der Benutzer hat die Möglichkeit, einen RSS-Feed deines bestimmten Nachhaltigkeitsberichtes (z.B. aus dem Jahr 2009) oder allen Nachhaltigkeitsberichten zu abonnieren, um über Änderungen in den jeweiligen Artikeln informiert zu werden
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer wählt den entsprechenden Link, um den RSS-Feed zu abonnieren.
Vorbedingungen	Es muss mindestens einen NB mit mindestens einem Artikel geben
Nachbedingung/Ergebnis	RSS-Feed ist abonniert
Standardablauf	
Alternative Ablaufschritte	-
Hinweise	Diese Funktion wird nur optional implementiert

Tabelle 9.56.: Anwendungsfall: „RSS-Feed abonnieren“

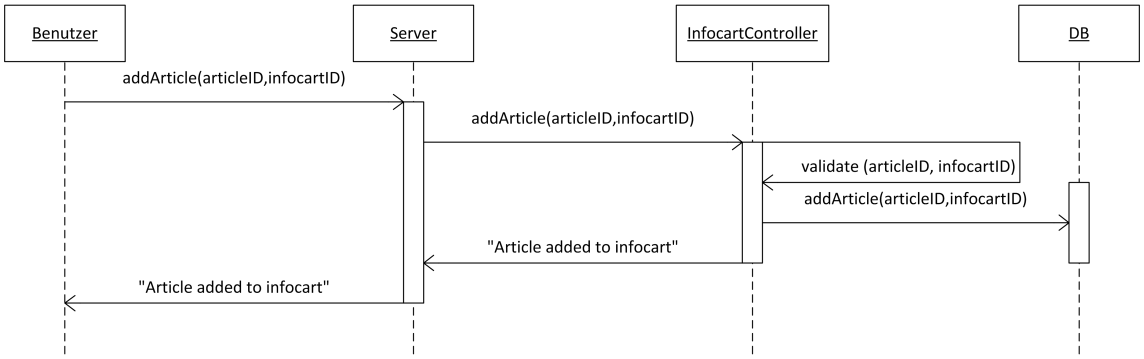

Abbildung 9.10.: Sequenzdiagramm „Artikel zum Infokorb hinzufügen“

9.4. Recommender Engine

In diesem Kapitel wird die Recommender Engine erläutert. Die Recommender Engine stellt die Implementierung des Vorschlagswesens dar. Auf Basis der protokollierten Daten können Aussagen darüber getroffen weredn, welche weiteren Artikel für einen bestimmten Benutzer interessant sein könnten. Diese Aussagen können vielseitiger Natur sein. Ein Vorschlag auf Basis der überhaupt am häufigsten betrachteten Artikel ist ebenso möglich, wie ein Vorschlag der interessanten Artikel für die Interessengruppe in der der Benutzer ist (siehe 9.1.2).
Um diese vielseitigen Vorschläge zu generierten besteht die Recommender Engine aus einzelnen funktionalen Bausteinen. Jeder Baustein besteht dabei aus einem Offline- und aus einem Online-Algorithmus. Der Offline-Algorithmus stellt die Datenmenge für die spezifische Engine bereit. Beispielsweise resultiuert aus der Offline-Engine zum Vorschlag „Artikel, die für deine Interessensgruppe interessant sind“ eine Auflistung welcher Artikel von welcher Interessensgruppe wie oft betrachtet wurde. Der Offline-Algorithmus wird läuft rein serverseit ab und kann vom Administrator des Systems gestartet manuell gestartet werden, auch ist es möglich dass der Administrator diesen Algorithmus in einem bestimmten Zeitabstand automatisch starten lässt. Hierzu wäre ein serverseitiges Shell-Skritp zu schreiben, welche die Offline-Algorithmen anstößt. Dieses Skript müsste die jeweiligen Update-Actions der einzelnen Controller aufrufen.
Der Online-Algorithmus, der im Gegensatz zum Offline-Algorithmus zur Laufzeit ausgeführt wird, selektiert aus der Ergebnismenge des Offline-Algorithmus die für dieses Szenario - ein spezifischer angemeldeter Benutzer - die passenden Vorschläge. Beispielsweise selektiert der Online-Algorithmus zum Vorschlag „Artikel, die für deine Interessensgruppe interessant sind“ aus der Ergebnismenge des Offline-Algorithmus die jeweilen Artikel, die für die Interessensgruppe, welcher sich der aktuell angemeldete Benutzer in seiner Profilverwaltung zugeordnet hat, interessant sind und am meisten betrachtet wurde, also entsprechend den höchsten Zähler haben. Anschließend wird jede Ergebnismenge des Online-Algorithmus der jeweiligen Vorschlagsfunktion vor Weitergabe zur Anzeige um die Schnittmenge der Ergebnismenge und der Menge der bisher von angemeldeten Benutzer betrachteten Artikel des Benutzers reduziert. Zu letztgenannter Aktivität werden die Ergebnisse der Protokollierung verwendet.
Im Folgenden werden die einzelnen Komponenten des Vorschlagswesens erläutert. Zuvor - um die Grundlage zu beschreiben - wird die Datenbasis beschrieben.

9.4.1. Datenbasis

Die in den folgenden Unterkapiteln vorgestellten Vorschlagsysteme ermitteln mittels gespeicherter Zugriffsdaten Vorschläge. Die Zugriffsdaten werden in dem Model AccessLog gespeichert. Pro Zugriff auf einen Artikel wird ein Eintrag erstellt. Das Access-Log enthält den Benutzer, den Artikel und einen Zeitstempel. Somit wird gespeichert, welcher Benutzer welchen Artikel wann betrachtet hat.

9.4.2. Am häufigsten betrachtete Artikel

Diese Vorschlagsart erstellt eine Liste von Artikel, welche nach der Häufigkeit der Zugriffe sortiert ist. Der Betrachtungszeitraum kann dabei variieren. Auch diese Vorschlagskomponente besteht er aus einem Online und Offline-Teil. Der Offline-Algorithmus ermitteln

auf welchen Artikel wie oft zugegriffen wurde. Per Systemkonfiguration kann der Systemadministrator der Organisation die STORM einsetzt einstellen, welche fünf Zeitfenster hierbei differenziert werden sollen. Beispielsweise können so Artikelzugriffe tages-, wochen-, monats- oder jahresspezifisch gespeichert werden. Die Anzahl der Zugriffe pro Zeitraum werden in den Parametern countPara[0..4] gespeichert. Die Parameter[0..4] beschreiben die Zeiträume. Der Offlineteil aktualisiert die Zugriffe auf Basis der Parameter. Die fünf Betrachtungszeiträume haben je einen Parameter. Dieser Parameter bestimmt den Betrachtungszeitrum in Tagen ab dem aktuellen Zeitpunkt. Neben den Betrachtungszeiträumen aktualisiert der Offlineteil die Zugriffe für den gesamten Zeitraum. Dazu wird ein Delta-Verfahren[3] benutzt, welches die Zugriffe seit dem letzten Durchlauf den bisherigen Summen der Zugriffe hinzufügt. Der letzte Durchlauf wird im Parameter lastRun gespeichert. Der Online-Algorithmus wählt aus dem Ergebnis des Offline-Algorithmus zur Laufzeit die relevanten Artikel die zur Anzeige kommen sollen. Dabei kann der Benutzer das Zeitfenster auswählen, und bekommt die am häufigsten in diesem Zeitraum betrachteten Artikel angezeigt. Der Onlineteil sortiert alle Elemente nach dem entsprechendern Counter-Attribut und gibt die spezifischen Artikeln weiter zur Visualisierung.
Gespeichert werden die Zugriffslisten in dem Model AccessArticleTyp. Diese legt pro Artikel einen Datensatz an und speichert folgende Attribute:

- Article: Den aufgerufenen Artikel
- countAll: Anzahl der Aufrufe
- countPara0: Anzahl der Aufrufe für den Betrachtungszeitraum des Parameters0
- countPara1: Anzahl der Aufrufe für den Betrachtungszeitraum des Parameters1
- countPara2: Anzahl der Aufrufe für den Betrachtungszeitraum des Parameters2
- countPara3: Anzahl der Aufrufe für den Betrachtungszeitraum des Parameters3
- countPara4: Anzahl der Aufrufe für den Betrachtungszeitraum des Parameters4

Der Parameter countAll speichert die Anzahl der Gesamtzugriffe und die countPara0 bis countPara5 für die einzelnen Betrachtungszeiträume.

9.4.3. Zuletzt betrachtete Artikel

Diese Vorschlagsart gibt die zuletzt betrachteten Artikel für einen bestimmen Benutzer wieder. Die Artikel werden ohne Mehrfachausgabe sortiert nach dem neusten Datum ausgegeben. Als Datengrundlagen dient hier das AccessLog - beschrieben in 9.4.1. Die stetige Protokollierung bildet die Offline-Komponente dieser Vorschlagsfunktion. Online werden aus dieser Datenbasis die Artikel selektiert, welche sich der angemeldete Benutzer als letztes angeschaut hat. Diese kommen zu Anzeige.

9.4.4. Ähnliche Artikel

In diesem Bereich geht es um ein Vorschlagsalgorithmus, der auf Basis der bisherigen Surfverhaltens des Benutzers - welche Artikel wurden bisher angeschaut - zu seinem Verhalten passende Artikel ermittelt und dem Benutzer diese vorschlägt. Dies wird realisiert indem eine Ähnlichkeit zwischen Artikeln ermittelt wird. Dieser Algorithmus wird unter dem

[3] Hierbei werden also nur die Änderungen seit dem letzten Durchlauf des Algorithmus benutzt. Die Steuerung dessen erfolgt über ein Zeitstempel.

Label „Item to Item Collaborative Filtering“ von Amazon eingesetzt. Dieser Algorithmus kann in der Kategorie *Search-based Methods* angesiedelt werden. Neben dieser Kategorie existieren weitere, wie zum Beisiel *Traditional Collaborative Filter* und *Cluster Models*. (Siehe [GL03], S. 1)

Nachdem in einem PG-Meeting die verschiedenen Kategorien vorgestellt wurden, hat sich die PG in Einklang mit den Betreuern für den „Item to Item Collaborative Filtering“-Algorithmus ausgesprochen. Die Vorteil dieses Algorithmus gehen aus einem Industry Report von Amazon.com hervor: „ [the algorithm] scales independently of the number of customers and number of items in the product catalog. [...] [The] algorithm produces recommendations in realtime, scales to massive data sets, and generates highquality recommendations.“([GL03], S. 1). Aufgrund dieser Entscheidung zu Gunsten des „Item to Item Collaborative Filtering“-Algorithmus wird auf die anderen Kategorien hier nicht weiter eingangen.

Der „Item to Item Collaborative Filtering“-Algorithmus wird der Kategorie *Search-based Methods* zugeordnet. „Search- or content-based methods treat the recommendations problem as a search for related items. Given the user's purchased and rated items, the algorithm constructs a search query to find other popular items by the same author, artist, or director, or with similar keywords or subjects. If a customer buys the Godfather DVD Collection, for example, the system might recommend other crime drama titles, other titles starring Marlon Brando, or other movies directed by Francis Ford Coppola“ ([GL03], S. 3). Die Item sind in diesem Anwendungsfall Artikel eines Nachhaltigkeitsberichts. Kunden stellen hier die Benutzer des Systems, beispielsweise Studenten, Mitarbeiter oder Professoren dar - diese können in einer Organisation natürlich vielfältig sein.

Der Algorithmus besteht aus einem Offline und einem Online Bereich. Offline - also unabhängig von der Laufzeit, serverseitig - werden die Ähnlichkeiten zwischen jeglicher Artikel-Artikel-Kombination auf Basis der Loggings ermittelt. Auf Basis dieser ermittelten Ähnlichkeit werden - online - in dem Moment in dem ein Benutzer einen Artikel betrachtet diesem Benutzer ähnliche Artikel vorgeschlagen. Der Offline-Part bildet so die für den Online-Part notwendige Datenbasis. Der Online-Part ermittelt zur Laufzeit - also in dem Moment in dem ein Benutzer einen Artikel konsumiert - die jeweils ähnlichen Artikel und zeigt diese unter dem aktuell betrachteten Artikel an.
Abbildung 9.11 zeigt die Funktionsweise des serverseitigen Offline-Algorithmus. Die Menge *product catalog* ist hierbei Menge aller Artikel auf die mindestens einmal zugegriffen wurde, die Datenbasis stellt hierbei das Logging dar. In Bezug auf jeden betrachteten Artikel (in der Abb. 9.11 als I_1 bezeichnet) wird der Konsument ermittelt. Die weiteren von diesem Konsumenten betrachteten Artikel (in der Abb. 9.11 als I_2 bezeichnet) werden ermittelt. Der Ähnlichkeitswert der beiden Artikel I_1 und I_2 wird nun erhöht, da diese beiden Artikel zusammen von einem Konsumenten gelesen wurden. Der Ähnlichkeitswert ist ein nummerischer Wert, der bei jeder Erhöhung inkrementiert wird.

Das Ergebnis des Offline-Algorithmus ist eine entsprechend gefüllte Datenbanktabelle mit folgenden Attributen:

- Artikel 1
- Artikel 2
- Ähnlichkeitswert

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by
       customer C
      Record that a customer purchased I1
       and I2
  For each item I2
    Compute the similarity between I1 and I2
```

Abbildung 9.11.: Item to Item Collaborative Filtering ([GL03], S. 4)

Die Vorteile dieses Offline-Algorithmus werden in dem *industry report* von Amazon.com so beschrieben: „The key to item-to-item collaborative filtering's scalability and performance is that it creates the expensive similar-items table offline. The algorithm's online component - looking up similar items for the user's purchases and ratings - scales independently of the catalog size or the total number of customers; it is dependent only on how many titles the user has purchased or rated. Thus, the algorithm is fast even for extremely large data sets. Because the algorithm recommends highly correlated similar items, recommendation quality is excellent." ([GL03], S. 4)
Der Online-Algorithmus wird ausgerufen, sobald ein Benutzer einen Artikel betrachtet. Für diesen Artikel werden die ähnlichsten Artikel ermittelt, also die Artikel-Artikel-Kombinationen die auf Basis des Ergebnisses des Offline-Algorithmus die jeweils höchsten Ähnlichkeitswert haben. Diese Artikel werden dem Benutzer als ähnliche Artikel direkt vorgeschlagen. Mit diesem „Item to Item Collaborative Filtering"-Algorithmus wird ein Vorschlagswesen realisiert, welches auf Basis des Surfverhaltens - welche Artikel betrachtet welcher Benutzer - Vorschläge generiert.

9.4.5. Artikel, die deine Follower betrachtet haben

Wie bereits im Kapitel 9.1.2 angesprochen, gibt es die Möglichkeit als Benutzer einen anderen Benutzer zu verfolgen, diesen anderen Benutzer als seiner Liste der Follower hinzuzufügen. Dazu wird in einer User-User-Relation ein entsprechender Eintrag getätigt. Diese Relation speichert den jeweiligen Benutzer, der gerade ein anderen Benutzer seiner Follower-Liste hinzufügt, und den Follower, also den Benutzer, der dieser Follower-Liste hiermit hinzugefügt wird. Diese User-User-Relation stellt die Grundlage für die Vorschlagsfunktion „Artikel, die deine Follower betrachtet haben" dar. Eine weitere Grundlage sind wiederrum die Logfiles. Auch bei dieser Vorschlagsfunktion existert ein Offline- und ein Online-Algorithmus. Der Offline-Algorithmus ermittelt auf Basis der Protokollierung - welcher Bentuzer hat sich welchen Artikel angeschaut - eine Auswertung folgender Natur: Zu jedem Benutzer wird gezählt wie oft sich dieser Benutzer einen Artikel angeschaut hat. Der Offline-Algorithmus füllt demnach eine Tabelle mit folgenden Parameter:

- User
- Artikel
- Counter

Auf dieser Datenbasis ermittelt der Online-Algorithmus on demand die Artikel, die deine Follower betrachtet haben. Es wird auf Basis des Sessionusers ermittelt, welche Follower dieser Benutzer hat. Anschließend wird die Menge der Artikel selektiert, die seine Follower betrachtet haben. Sortiert nach dem Counter in absteigender Reihenfolge wird die Menge der Artikel, die deine Follower betrachtet haben, als Ergebnismenge entsprechend zurückgegeben und kann auf der entsprechenden Seite dargestellt werden.

9.4.6. Artikel, die für deine Interessengruppe interessant sind

Zu einem Benutzer wird das Attribut „Interessengruppe" gespeichert, genaueres siehe Abschnitt 9.1.2. Ein denkbares Szenario an einer Universität könnten die Interessengruppen Student, Mitarbeiter und Professor darstellen. Um für Interessensgruppen gezeilte Vorschläge zu ermitteln, ist es grundlegend natürlich notwendig das der jeweilige Benutzer das Attribut pflegt. Weiterhin ist auch hier ein zweigeteilter Algorithmus notwendig:

ein Offline- und ein Online-Algorithmus. Durch den Offline-Algorithmus wird die Datenbasis für diese Vorschlagsfunktion generiert. Die Basis stellt hierbei wieder das Logfile dar: es wird zu jedem Artikel-Aufruf der Artikel und der Benutzer protokolliert. Der Offline-Algorithmus bildet auf dieser Grundlage eine Auswertung darüber, welcher Artikel von welcher Interessensgruppe wie oft aufgerufen wurde. Die Ergebnismenge des Offline-Algorithmus besitzt demnach folgende Parameter:

- Interessengruppe
- Artikel
- Counter

Auf diesem Ergenis setzt der Online-Algorithmus auf. On demand wird ermittelt, welche Interessensgruppe der Sessionuser inne hat - sofern diese gepflegt ist. Zu dieser Interessensgruppe werden die Artikel mit dem höchsten Counter selektiert und entsprechend zur Anzeige gebracht.

9.5. Weitere Features

Um eine nachvollziehbare und vereinfachte Benutzung mit Blick auf Dialogorientierung und -förderung zu gewährleisten, bietet Storm folgende, weitere Features:

- Tag-Cloud
- Autovervollständigung
- Sprachauswahl
  - serverseitig
  - clientseitig
- News
- Social Bookmarks/ Networks
- Add to Friends
- Suche
- Hilfe

9.5.1. TagCloud

In STORM ist eine Schlagwort-Wolke (TagCloud) integriert. Der Funktionsablauf dieser Tag-Cloud ist wie folgt:
Ein User gibt in der Anwendung zu einem Artikel ein bestimmtes Schlagwort ein. Diese werden vom Grails - Plugin Taggable empfangen und in der Datenbank abgelegt. Auf Grundlage der so gesammelten und in der Datenbank abgespeicherten Daten analysiert das Grails-Plugin Tag Cloud die Schlagwörter nach Relevanz und Häufigkeit aus. Nachdem die Analyse durch das Tag Cloud Plugin erfolgt ist, werden die wichtigsten Schlagwörter innerhalb der View mittels eines HTML - Bereiches auf der Startseite ausgegeben. Durch dieses Verfahren erfolgt zugleich auch die Dynamisierung der Schlagwort-Wolke, da eine Veränderung an den Schlagwörtern auch immer zu einem veränderten Aussehen der Schlagwort-Wolke führt. Die Ausgabe der Daten erfolgt innerhalb der View wie gesagt als HTML -

Bereich. Dieser wird zur Unterstützung des Benutzers mit Hilfe einer JavaScript-Bibliothek optisch verändert. Häufige Schlagwörter werden hierdurch in einer höheren Schriftgröße als weniger häufige Begriffe angezeigt. Die JavaScript-Bibliothek, die hierfür genutzt wird lautet JQuery (siehe (http://www.jquery.com/)). jQuery zeichnet sich insbesondere durch seine hohe Performanz aus und wurde gewählt um den User nicht zu lange auf die Startseite warten zu lassen (zur Performance siehe (http://www.ajaxschmiede.de/jquery/geschwindigkeitsvergleich-der-javascript-bibliotheken/)).

Technischer Hintergrund:
Sobald der Benutzer die Anwendung aufruft, sieht er auf der Startseite die Tag-Cloud. Die Tags, die von der Mehrzahl der Benutzer vergeben wurden, werden vom Grails PlugIn Tag Cloud aus dem Model geholt und an die View weitergegeben.

9.5.2. Autovervollständigung

STORM bietet eine Autovervollständigung, die es dem Benutzer ermöglicht eine Auswahl von vermutlichen Begriffen einzugeben. Während dieser Eingaben in ein Suchfeld macht, bekommt er so die Möglichkeiten, wie das aktuelle Wort vervollständigt werden kann und welche Worte seine Suche zusätzlich verbessern können.

Für die Autovervollständigung wird eine Funktionalität des RichUI - Plugin genutzt, welches bereits zur Anzeige von Eingabebereichen innerhalb des Administrationsbereiches benutzt wird. Durch dieses Plugin kann die Eingabe des Benutzers ständig mit den Tabellen in der Datenbank verglichen werden. Die Daten, die aus der Datenbank zum Vergleich anwandt werden, beinhalten die Namen von Nachhaltigkeitsberichten, Artikeln und Kommentaren, sowie deren Inhalte.

Technischer Hintergrund:
Sobald der Benutzer einen Begriff in einem Suchfeld eingibt, wird das RichUI-Plugin der Anwendung aufgerufen. Alle Begriffe die dem gesuchten ähnlich sind werden dem Anwender sodann in einer Vorschlagsliste unterhalb des Suchfeldes aufgezeigt. Dem Anwender bleibt sodann die Wahl, ob er den Vorschlag annehmen möchte oder seine Eingabe vervollständigen möchte.

9.5.3. Sprachauswahl

Die Anwendung ist in mehreren Sprachen verfügbar, das bezieht sich auf die Beschriftung von Buttons, Menüeinträgen, Hilfetexten etc., nicht aber auf Inhalte eines Nachhaltigkeitsberichtes. Diese sind nur in der Sprache angezeigt, in der sie geschrieben wurden.
In der fertigen Version der Anwendung sind Deutsch und Englisch angeboten, weitere Sprachen können modular hinzugefügt werden. Das Umschalten der Sprachen funktioniert einfach mittels Auswahl von einem Dropdown-Feld in dem, die vorhandenen Sprachen aufgelistet sind. Jede Sprache wird in der eigenen, sprachtypischen Schrift, präsentiert.
Alle benötigten Bezeichnungen pro Sprache sind in einer XML-Datei gespeichert. Für das Einbinden der Worte muss die XML-Datei in der Sektion der jeweiligen Sprache geparst werden.
Es muss unterschieden werden zwischen der Standardsprache der Anwendung, und der Sprache die der Benutzer ausgewählt hat. Die Standardsprache der Anwendung wird in STORM selbst gespeichert und die Sprache des Benutzers, soweit sie von der Standardsprache der Anwendung abweicht, wird clientseitig in einem Cookie gespeichert.

Name	Sprache auswählen
Beschreibung	Der Benutzer hat die Möglichkeit, eine Sprache (Deutsch/Englisch) auszuwählen.
Beteiligte Akteure	Benutzer, System
Auslöser	Benutzer möchte eine (andere) Sprache auswählen und wählt dazu den entsprechenden Link
Vorbedingungen	Es muss mindestens zwei Sprache zur Auswahl geben.
Nachbedingung/Ergebnis	Die gewünschte Sprache wird auf Webseite angezeigt
Standardablauf	
Alternative Ablaufschritte	-
Hinweise	

Tabelle 9.57.: Anwendungsfall: „Sprache auswählen“

9.5.4. Template-Auswahl

Es besteht die Möglichkeit das Template des Systems zu wechseln, und damit das Erscheinungsbild der Anwendung zu ändern. Standardmäßig wird ein default-Template ausgeliefert. Nachdem ein neues Template erstellt wurde (vgl. hierzu Dokumentation zum Template-System) kann dieses im Administrationsmenü aktiviert werden. Dazu muss der Name des Templates angegeben werden. Es kann immer nur ein Template parallel aktiviert sein. Sollten mehrere Templates aktiviert worden sein, wird immer das erste aktivierte Template verwendet.

9.5.5. RSS-Import

Um externe RSS-Feeds (z.B. News) in die Anwendung zu integrieren, besteht die Möglichkeit einen oder mehrere RSS-Feeds zu importieren. Die Konfiguration von RSS-Feeds erfolgt im Administrationsmenü. Die Darstellung erfolgt standardmäßig auf der Startseite. Um einen RSS-Feed importieren zu können muss dessen URL angegeben werden. Zusätzlich kann definiert werden, ob der Titel, der Autor, die Beschreibung und das Datum ausgegeben werden sollen. Über einen booleschen Schalter kann der RSS-Feed aktivier bzw. deaktiviert werden. Darüber hinaus kann die Anzahl der anzuzeigenden Einträge und die maximale Länge der darzustellenden Beschreibung parametrisiert werden. Über einen weiteren booleschen Parameter kann festgelegt werden ob der Link im aktuellen oder in einem neuen Fenster aufgerufen werden soll.

9.5.6. Newsletter

Um dem Benutzer die Möglichkeit zu geben, sich aus der Anwendung heraus für einen Newsletter anzumelden, können externe Newsletter-Systeme integriert werden. Die Konfiguration erfolgt im Administrationsbereich der Anwendung. Hierbei können allgemeine Angaben zum Newsletter hinterlegt werden (Provider, Beschreibung), darüber hinaus muss die Formular-Action und die Felder des externen Newsletter-Anbieters angegeben werden. Die Felder sind der Name, Vorname und die Email-Adresse. Anhand dieser Angaben wird das Formular des externen Newsletter-Anbieters in der Anwendung reproduziert. Sollte ein Nutzer bereits am System angemeldet sein, werden die Felder entsprechend vorbelegt,

sodass eine Anmeldung an einem Newsletter System mit nur einem Klick möglich ist. Es können beliebig viele externe Newsletter Systeme eingebunden werden.

9.5.7. Fragen-System

Das Fragen-System der Anwendung bietet dem Benutzer die Möglichkeit Fragen zu stellen, Fragen zu kommentieren und Antworten zu bestehenden Fragen zu lesen. Das System ist dem Prinzip von „Direktzu“ nachempfunden [Dir09]. Die einzelnen Fragen werden im Administrationsbereich der Anwendung verwaltet. Zu jeder Frage muss ein Titel sowie die eigentliche Frage eingegeben werden. Darüberhinaus kann ein Benutzer mit entsprechenden Rechten eine Antwort zu einer Frage einstellen (z.B. Präsident der Universität). Über eine boolesche Variable kann die Frage mit einer entsprechenden Antwort schließlich freigeschaltet werden. Der Benutzer der Anwendung sieht immer eine aktuelle Liste aller neuen Fragen. Daraufhin kann er einzelne Fragen bewerten. In einer weiteren Liste werden die Fragen der Bewertung nach sortiert angezeigt. Die am besten bewerteten Fragen können/sollten daraufhin von einem Benutzer mit entsprechenden Rechten beantwortet werden. Alle beantworteten Fragen werden in einer Dritten Liste dem Datum nach sortiert ausgegeben. Schließlich besteht die Möglichkeit eigene Fragen mit Titel und Fragetext einzugeben. Diese kann daraufhin bewertet und beantwortet werden.

9.5.8. Hilfe-System

Damit der neue Anwender bei der Bedienung des Systems ausreichend unterstützt werden kann, ist eine Hilfefunktion implementiert. Diese Hilfe unterstützt den Anwender indem zu jedem beliebigen Teil des Systems Hilfetexte vom Administrator angelegt werden können. Diese sind dann für alle Anwender verfügbar und können betrachtet werden. Das Hilfesystem selbst ist in Themen und Artikel aufgeteilt, wobei verschiedene Artikel einem Thema zugeordnet sind. Der grundsätzliche Aufbau ist allgemein bekannten Nutzeranwendungen wie Microsoft Office oder OpenOffice nachempfunden, so dass der Anwender keine Einarbeitungszeit benötigt. Für die Pflege der Hilfethemen ist der Aministrator verantwortlich. Nur dieser besitzt die Berechtigung neue Themen und neue Artikel anzulegen und Inhalte zu pflegen.

9.5.9. News

Es gibt eine Sektion für Neuigkeiten (News), in der redaktionell auf Änderungen hingewiesen wird. Dies kann, eine News zur Fertigstellung eines neuen Nachhaltigkeitsberichtes oder auch Neuigkeiten über die Institution, welche die STORM verwendet, sein.

9.5.10. Social Bookmarks/Networks

Innerhalb der Anwendung ist es dem Benutzer ermöglicht, die Applikation auf Social Bookmark Seiten (Delicious, Mister Wong, etc.) einzutragen. Der Benutzer ist dazu auf die Seite eines Dienstes (pro Dienst ein Link, Linkbezeichnung ist ein Icon des Dienstes) in einem neuen Browserfenster weitergeleitet. Auf der Seite des Dienstes ist der Benutzer nach erfolgreichem Login (beim Dienstanbieter) auf eine Seite geführt, auf der er die Applikation (STORM) bewerten kann. Des Weiteren ist es dem Benutzer ermöglicht einzelne Artikel oder Berichte auf Social Network Seiten (Facebook, Twitter, etc.) einzutragen.

Auch bei Social Networks ist der Benutzer, analog zu Sozial Bookmarks, in einem neuen Browserfenster weitergeleitet. Auf der Seite des Dienstes wird der Benutzer nach erfolgreichem Login (beim Dienstanbieter) auf eine Seite geführt, auf der er die Verlikung zu dem Artikel oder Bericht mit einem Kommentar versehen und auf der Social Network Seite veröffentlichen und somit anderen zu Verfügung stellen kann.

Technischer Hintergrund:
Beim Aufbauen der View werden die aktiven Social Bookmark, -Network Seiten (solche, die im System eingetragen und als aktiv markiert sind) dargestellt. Bei Änderungen an den aktiven Diensten wird die View entsprechend vom Controller aktualisiert.

9.5.11. Startseite

Nach dem Abschluss einer erfolgreichen Anmeldung am System wird die Startseite angezeigt. Die Startseite kann ebenfalls mit einem Klick auf das STORM Logo in der Software erreicht werden. Aus der Startseite sollen die einzelnen Funktionsbausteine der Recommender Engine (siehe 9.4) integriert werden. Des weiteren gibt die Startseite in Form eines einleitenden Textes - der wiederrum von jeder die Software STORM einsetzenden Organisation adaptiert werden kann - einen allgemeine Übersicht. Auch RSS-Feeds werden auf die Startseite integriert. So gibt diese Startseite dem Benutzer einen guten Überblick und läd zum Leser der vorgeschlagenen Artikel ein.

Teil IV.

Projektabschluss

10. Testbericht

Dieser Bericht beschreibt im Detail, wie die Anwendung STORM getestet wurde.

10.1. Warum muss eine Webanwendung getestet werden?

Grundsätzlich betrachtet erfüllt eine Webanwendung Funktionen genau wie auch andere Software dies tut. Es gibt daher keinen Grund, warum eine Webanwendung nicht auch die gleichen Qualitätsmerkmale aufweisen sollte, im Gegenteil, durch ihre Art der Nutzung und Verbreitung in einem Webbrowser muss eine Webanwendung sogar weitere Qualitätsmerkmale erfüllen. Zu diesen gehören unter anderen Erreichbarkeit, Verfügbarkeit, Fehlertoleranz und Sicherheit, siehe auch Seminararbeit *Qualitätssicherung in der Softwareentwicklung mit Fokus auf Web-Applikationen* TODO: In der fertigen Version bitte den Verweis einkommentier
Daher wurde die Anwendung STORM ausführlich getestet, wie wird in den folgenden Kapiteln beschrieben.

10.2. Testverfahren

Der Test verläuft in zwei Phasen, Phase 1 besteht aus Unit- und Integration Tests, diese werden von jeder Entwicklergruppe nach eigenem Ermessen und in Eigenverantwortung durchgeführt.
Phase 2 besteht aus einem Test der Anwendung zur Laufzeit, d.h. alle möglichen Aktionen auf der Webseite werden getestet und das Ergebnis dokumentiert.
Zur Organisation wird das Verfolgungssystem aus der eGroupware benutzt: https://vittel.informatik.uni-oldenburg.de/egroupware/index.php?menuaction=tracker.tracker_ui.index
Fehlerhaftes Verhalten der Anwendung wird in diesem System dokumentiert und zur Bearbeitung an die entsprechenen Personen/Gruppen delegiert.

10.2.1. Unit-und Integration Tests

Die Unit- und Integration Tests führt jede Gruppe in Eigenverantwortung und nach eigenem Ermessen durch. Dabei handelt es sich um *White-Box-Tests*, d.h. die Testenden kennen den Aufbau und Ablauf der getesteten Funktion, sowie die erwartete Ausgabe zu einer Eingabe.
Dazu werden in Grails spezielle Tests geschrieben, zum Aufbau und Durchführen dieser Tests siehe http://grails.org/doc/latest/guide/9.%20Testing.html.

10.2.2. Laufzeittests

Die Laufzeittests sind vollständige *Black-Box-Tests*, d.h. der Testende kennt die Eingabe und die erwartete Ausgabe, nicht jedoch die internen Strukturen der getesteten Funktionen. Um eine Liste der zu testenden Funktionen zu erstellen, werden die Anforderungen an das System unter Zuhilfenahme der Anwendungsfalldiagramme analysiert und daraus die Testfälle erstellt. Beispiel: Es existiert die Anforderung: Ein anonymer Nutzer kann sich registrieren und sich danach einloggen. Daraus ergibt sich der Testfall, dass ein anonymer, unregistrierter Nutzer sich nicht einloggen kann, nach einer erfolgreichen Registrierung aber schon. Der Test läuft dann folgendermaßen ab:
Der Test-Nutzer ruft die Weboberfäche zu STORM auf und klickt auf einloggen (Eingabe). Das System verweigert dies (erwartete Ausgabe). Der Nutzer registriert sich (Eingabe), das System teilt dem Nutzer den Erfolg mit (erwartete Ausgabe). Der Nutzer loggt sich ein (Eingabe), das System begrüßt den nun eingeloggten Nutzer (erwartete Ausgabe).
Verhält sich die Anwendung wie erwartet, so ist der Test bestanden, verhält sie sich aber anders, wird der Fehler dokumentiert und zur Bearbeitung delegiert (in der eGroupware). Mittels dieses Verfahrens wurde die Anwendung STORM während der gesamten Entwicklungszeit kontinuierlich getestet, wobei acht verschiedene Entwicklungsstände als Meilensteine gesondert getestet und dokumentiert wurden. Siehe 10.3.
Die Summe der so ermittelten Testfälle ergibt ein Testszenario, das einen Überblick über alle zu testenden Funktionen schafft. Zu den ersten sieben Meilensteinen wurden jeweils Teile dieses Szenarios getestet, zum letzten Meilenstein wurde das komplette Szenario durchgespielt. Der Bericht zu diesem kompletten Test ist in Abschnitt 10.3.8 nachzulesen. Das Szenario ist wie folgt aufgebaut:

10.2.2.1. Testszenario

Im Rahmen dieses umfassenden - sogenannten großen - Szenarios werden neben allen Funktionen der Software auch die jeweiligen Auswirkungen der Funktionen getestet. Dieses „große“ Szanrio ist in der Hinsicht groß, als das es alle möglichen Funktionen der Software STORM umfasst. So kann zum Beispiel die Änderung eines Parameters bei einer Funktion das Ergebnis beeinflussen, entsprechend müssen die Tests ganzheitlich und allumfassend gestaltet sein. Inkludiert in der Beschreibung des großen Testszanrios wird hier auch ein Überlick über die verschiedenen Bereiche gegeben.

Für das Testszenario gehen wir von folgendem Umfeld aus: Die Software STORM wird von einer Organisation „gefunden“. Die Mitglieder der Organisation probieren das Web 2.0 Frontend aus. Anschließend registriert sich der Nutzer und wendet die Funktionen des Backends die einem normalen Nutzer zur Verfügung stehen an. Der Nutzer bekommt dann die Rolle des Redaktuers zugeordnet und testet die ihm nun verfügbaren Funktionen des Eingabe Editors. Zuletzt bekommt der Nutzer administrative Rechte und wendet die Funktionen des Administrationsbereichs an.

Als anonymer Nutzer ist man im Frontend aktiv. Den Verwaltungsbereich - das Backed - erreicht man nur als registrierter Benutzer. Folgend wird das Testszenario aufgeteilt in die verschiedenen Entwicklergruppen beschrieben. Wobei nur die erste Entwickergruppe - Web 2.0 - auch im Frontentbereich aktiv ist.

Um das Szenario anschaulicher zu gestalten, wird angenommen, Herr Müller und Herr Meier befassen sich mit der Software STORM. Sie sind Mitarbeiter einer Universität und prüfen STORM auf Einsatztauglichkeit für die Universität.

Szenarioteil Web 2.0

Herr Müller installiert die Anwendung STORM inklusive eines Standarddatensatzes und kann diese nun ausprobieren ohne eingeloggt zu sein. Folgende Funktionen stehen ihm zur Verfügung:

- Suchfunktion
  - Artikel suchen: Eingabe → Liste der gesuchte Artikel angezeigt
- Helpsystem
  - Helparticle suchen : Eingabe → Liste der gesuchte Artikel angezeigt
  - Helparticle anzeigen: Topic-name auswählen → Artikel anzeigen

Herr Müller kann sich auch registrieren und somit stehen ihm jetzt mehr Funktionen zur Verfügung.

- Nach dem Einloggen:
  - Helpsystem
  - Helparticle anlegen: Eingabe → Helparticle anlegen
  - Helparticle bearbeiten: Eingabe → Helparticle anlegen
  - Helparticle anzeigen: Liste der Helparticle → Article-Name auswhälen
  - Helparticle anzeigen: Liste der Helparticle → Article-Name auswhälen
  - Helparticle löschen: Liste der Helparticle → Article löschen

Infokorb

- Infokorb → Infokorb anzeigen
- Infokorb → Infokorb anzeigen → Infokorb neu anlegen
- Infokorb → Inhalt auf einer Seite
- Infokorb → Inhalt als PDF
- Infokorb → Meine Infokörbe
- Infokorb → Meine Infokörbe → Laden, Bearbeiten, Löschen [jeweils Symbole]
- Infokorb → Infokörbe anderer Benutzer
- Infokorb → Infokorb anzeigen → [Artikelname] Infokorb → Infokörbe anderer
- Benutzer → Benutzer [Benutzername] (Infos zu dem Benutzer) [Nur Admin]

Hilfe

- Hilfe → Hilfethema anlegen [Nur Admin]
- Hilfe → Hilfethemen verwalten → bearbeiten, löschen [Nur Admin]
- Hilfe → Hilfeartikel anlegen [Nur Admin]
- Hilfe → Hilfeartikel verwalten → bearbeiten, löschen [Nur Admin]
- Hilfe → Übersicht
- Hilfe → Suchen

Kommentar

- Artikelansicht → Kommentarartitel anlegen → Inhalt schreiben → Kommentar speichern Nur angemeldeter User
- Artikelansicht → [Kommentar] → antworten Nur angemeldeter User
- Artikelansicht → [Kommentar] → löschen Nur Admin

Bewertung

- Artikelansicht → 5-Sternebewertungssystem nutzen
- Kommentaransicht → Daumen-Bewertungssystem nutzen Nur angemeldeter User, ausser Autor des Kommentars
- Berichte werden automatisch durch Bewertung der Artikel des Berichts bewertet

Suchen

- Suchbegriff in Textfeld neben Drop-Down-Menü für Suchfilter eingeben → Filter auswählen → ausführen
- Hilfe → Suchen

Social Networks

- Artikelansicht → Icon aus Social-Networks-Leiste auswählen

Szenarioteil Systemadministration

Herr Müller beauftragt Herrn Meier die Administration zu übernehmen, dieser nutzt dazu den Admin-Account von STORM und kann somit diverse Einstellungen verwalten. Diese Einstellungen sind die Folgenden:

- User
  - User registrieren → Startseite → Registrieren
  - User Password vergessen → Startseite → Passwort vergessen
  - User Anmelden → Startseite → Anmelden
  - User abmelden → Startseite → Abmelden
  - User Meine Daten ändern → Startseite → Anmelden → Meine Daten → Mein Profil
  - User Meine Daten letze Aktionen → Startseite → Anmelden → Meine Daten → Meine letzen Aktionen
  - User Meine Daten letze Vorschläge → Startseite → Anmelden → Meine Daten → Meine letzen vorschläge
- Startseite:
  - Die verschiedenen Infos abrufen
- Als Admin anmelden:
  - Im Admin Menu „ADMIN“:

        * User bearbeiten→ CRUD
        * Rolle bearbeiten→ CRUD
        * Rechte bearbeiten→ CRUD
        * Module bearbeiten→ CRUD
    - Im Admin Menu Logs:
        * die verschiedenen Logs anschauen

Szenarioteil Schema Editor

Herr Meier befördert Herrn Müller zum Redakteur, dies ist eine vom Admin gepflegte Rolle, die es erlaubt, Schemata und Berichte (siehe Kapitel 10.2.2.1) zu erstellen und zu verwalten. Herr Müller hat jetzt viele neue Funktionen zur Verfügung.

- Artikel als PDF anzeigen
    - Startseite → NB wählen → Artikel auswählen → PDF (Icon) (Voraussetzung: Artikel wurde freigegeben, d.h. PDF wurde generiert)
- Infokorb als PDF anzeigen
    - Startseite → Infokorb → PDF (Icon)
- NB als Pdf anzeigen
    - Startseite → NB wählen → PDF (Icon)
- Diagramm anzeigen
    - (von egal woher) → Klick auf Indicator zeigt Diagramm
    - Klick auf Indicator zeigt Diagramm → dadrunter Auswahl der Jahre (von bis) → Diagramm aktualisieren
- Schema
    - Schema anlegen
    - Schemata Verwalten → Schemaliste exportieren (CSV, PDF, XLS,...)
    - Schemata Verwalten → Liste durchsuchen/Filtern
    - Schemata Verwalten → Aktion wählen (dropdown)
    - Schemata Verwalten → Schema auswählen
    - Schemata Verwalten → Schema auswählen → Schema dteails bearbeiten
    - Schemata Verwalten → Schema auswählen → Schema löschen
    - Schemata Verwalten → Schema auswählen → Indikatoren in diesem Schema → Liste exportieren (CSV, PDF, XLS,...)
    - Alle Indikatoren im SE-Bereich sind anklickbar und führen zu /show
    - Schemata Verwalten → Schema auswählen → Ebene auswählen → Ebene löschen
    - Schemata Verwalten → Schema auswählen → Ebene auswählen → Ebene bearbeiten

  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Neue Ebene
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Indikatoren diese Ebene → Exportieren (csv, pdf, xls,...)
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Indikatoren → Indikator hinzufügen
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Indikatoren → Indikator entfernen
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Indikatoren → Indikatoren Filtern
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Indikatoren → Listen exportieren (globale/lokale Indikatoren)
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Schema importieren
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Schema importieren → Undo
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Verschieben
  - Schemata Verwalten → Schema auswählen → Ebene auswählen → Übersicht

- Kategorie:
  - Kategorie anlegen
  - Kategorien Verwalten → Kategorie auswählen
  - Kategorien Verwalten → klick auf eine Kategorie → anzeigen
  - Kategorien Verwalten → klick auf eine Kategorie → bearbeiten
  - Kategorien Verwalten → klick auf eine Kategorie → löschen
  - Kategorien Verwalten → klick auf eine Kategorie → Liste der Indikatoren zu dieser Kategorie

- Indikatoren:
  - Indikatoren anlegen
  - Indikator anlegen → Indikatorliste
  - Verlauf von Indikatoren: überall wo Indikatoren verlinkt sind, die in mehr als einem NB benutzt werden, sollte hinter dem Indikator ein Verlaufspfeil stehen (nach oben bzw. nach unten), welcher den Verlauf zum letzten Jahr deutlich macht
  - Indikatoren Verwalten → Indikator anlegen
  - Indikatoren Verwalten → Liste exportierne (csv, pdf, xls,..)
  - Indikatoren Verwalten → Liste durchsuchen/Filtern
  - Indikatoren Verwalten → Aktion wählen (dropdown)
  - Indikatoren Verwalten → Indikator auswählen → hier werden weitere Indikatoren angezeigt, die einen Bezug zu diesem Indikator haben und den Benutzer auch interessieren könnten

  - Indikatoren Verwalten → Indikator auswählen → Einzelne Bereiche bearbeiten
  - Indikatoren Verwalten → Indikator auswählen → Diagramm bearbeiten
  - Indikatoren Verwalten → Indikator auswählen → Indikator details über das Icon
  - Indikatoren Verwalten → Indikator auswählen → Indikator löschen
- Validierung
  - Eingabe → Berichte Verwalten → Bericht über Name anklicken → Validieren
  - Eingabe → Berichte Verwalten → Liste der Artikel (Icon) → Artikel wählen → Bearbeiten → Validierung oben → Falls nicht valide, Indikatoren hinzufügen
  - Eingabe → Berichte Verwalten → Bericht über Name anklicken → Validieren → Listen exportieren (csv, pdf, xls,...)
- XML
  - http://localhost:8080/Storm/article/xml/<ID> (ID, z.b. 1)
  - http://localhost:8080/Storm/report/xml/<ID> (ID, z.b. 1)
  - http://localhost:8080/Storm/indicator/xml/<ID> (ID, z.b. 1)
  - http://localhost:8080/Storm/TNode/xml/<ID> (ID, z.b. 1)
  - http://localhost:8080/Storm/TReport/xml/<ID> (ID, z.b. 1)
- RSS-Feeds
  - http://localhost:8080/Storm/feed/feed

Szenarioteil Eingabe Editor

Mit der Rolle Redakteur hat Herr Müller jetzt auch die Möglichkeit, den Eingabe Editor und die Mediathek zu benutzen.

- Eingabeeditor:
  - Hauptmenü Bericht verwalten
    - nach Bericht suchen (mit und ohne Filter)
    - Bericht erstellen/ansehen/editieren/löschen
  - Hauptmenü Artikel verwalten
    - nach Artikel suchen (mit und ohne Filter)
    - Artikel erstellen/ansehen/editieren/löschen
- Mediathek:
  - Hauptmenü Datei hochladen
    - lokale Datei hochladen
    - Datei online einbinden
  - Hauptmenü Dateien verwalten
    - nach Dateien suchen (mit und ohne Filter)

- Auflistung aller Attribute
- Auflistung aller Einsatzorte der Datei

– Article Editor

* Einfügen von Bildern
* Einfügen von Videos

10.2.2.2. Selenium

Um eine Webanwendung mit allen Funktionen zu testen, werden diese Tests üblicherweise automatisiert. Für die Tests von STORM wurde die Testsuite *Selenium* genutzt (http://seleniumhq.org/). Diese Testsuite ist als Plugin für Grails verfügbar (http://www.grails.org/plugin/selenium), ausserdem existiert eine IDE zum Erstellen der Tests (http://seleniumhq.org/projects/ide/). Für das Testen von STORM wurde die Selenium IDE genutzt. Die IDE unterstützt beim Erstellen und Ausführen der Tests, kann aber nicht alle Arbeiten abnehmen, da nicht alle Script-Befehle über die IDE verfügbar sind. Weiterhin können nicht alle Funktionen vollständig automatisch getestet werden, der Grund dafür sind bestehende Sicherheitsrichtlinien bei JavaScript, das von der Selenium Testsuite verwendet wird.
Die IDE unterstützt als Browser ausschließlich Firefox, in allen anderen Browsern ist derzeit das Testen von HTML Input Elementen des Typs *file* aus Sicherheitsgründen nicht möglich. Zukünftige Versionen von Selenium sollen dies beheben.
Desweiteren ist das Auffinden von Elementen in einer Webseite nicht absolut fehlerfrei. Es gibt verschiedene Möglichkeiten, die von verschiedenen Browsern unterschiedlich gut unterstützt werden. Dazu gehören unter anderem das Auffinden über *XPath*, *DOM*, *Hyperlink Link Text*, *Name* und *ID*, diese Möglichkeiten wurden für das Testen von STORM mittels Selenium genutzt. Weitere Informationen unter http://seleniumhq.org/docs/04_selenese_commands.html#locating-elements
Hinweis: Für den Browser *Internet Explorer* müssen einige Tests u.U. angepasst werden, da vor allem XPath nicht ausreichend unterstützt wird.
Mit Selenium kann man Aktionen erzeugen, jede Aktion spiegelt eine Tastatureingabe (einzelnes Zeichen bis zu einem ganzen Satz) oder einen Mausklick (Klick auf einen Link, auf einen Button, auf eine Checkbox usw.) wieder, oder verifiziert die Anwesenheit und Beschaffenheit eines Elementes. Ein solches Element kann beliebiger Inhalt einer gültigen Webseite sein (Text, Link, DIV-Container usw.).
Solche Aktionen lassen sich zu einem Testfall zusammenfassen, so dass z.B. ein kompletter Anwendungsfall in einem Testfall zusammengefasst wird. Testfälle wiederum lassen sich in einer Testsuite zusammenfassen. Eine Übersicht der vorhandenen Testsuiten- und Fälle gibt es in Kapitel 10.3.8. Es wurden insgesamt 7 Testuiten mit ca. 100 Testfällen erstellt, die über 1900 Aktionen durchführen.
Die Tests sind im SVN vorhanden: /code/Trunk/Storm/web-app/Selenium
Um die Tests durchzuführen, muss die Selenium IDE installiert sein. Einzelne Testfälle können mit *Datei öffnen* geöffnet und dann gestartet werden, eine Testsuite wird über *Test Suite Öffnen* geöffnet. Zu beachten ist, dass die Testfälle im Ordner *tests/01Mediathek* absolute Pfadangaben enthalten, diese müssen auf jedem System angepasst werden. Relative Pfadangaben sind bei diesen speziellen Tests nicht möglich, da Datei-Uploads und Downloads getestet werden.
Desweiteren beziehen sich viele der Aktionen auf bestimmten Inhalt, so dass die Tests mit

den eingepflegten Testdaten durchgeführt werden müssen. Die Daten der SVN-Version 2623 TODO: aktuelle Version einfügen nach letztem Durchlauf der Grails Development Umgebung genügen für einen vollständigen Testdurchlauf. Auch muss das System „frisch“ sein, da an vielen Stellen geprüft wird, ob z.B. ein Inhalt korrekt gelöscht wurde, wenn aber von anderer Stelle der gleiche Inhalt generiert wurde, schlägt der Test fehl, obwohl alles korrekt ist.
Die Tests lassen sich in unterschiedlichen Geschwindigkeiten abspielen, es empfiehlt sich, die niedrigste Geschwindigkeit zu wählen, da sonst die IDE oftmals ein HTML-Element ansteuert, das noch gar nicht geladen wurde.
Unter diesen Voraussetzungen lässt sich die Test-Suite TODO: name einfügen laden und durchführen.

10.2.3. Verfolgungssystem

Anleitung zur Bedienung des Bug Tracking Systems.
Das System erlaubt die übersichtliche Darstellung von Fehlverhalten (Bugs), dabei kann unter vielen verschiedenen Ansichten gewählt werden. Es gibt die Verfolgungsqueue, diese gilt als Richtlinie der Deadline zum Beheben eines Bugs.
Auswahlmöglichkeiten:

- alle
- Alpha
- Beta
- 28. Juli
- 31. Juli
- 31. August
- final Release
- Feature Requests

Die Kategorie unterteilt das Ticket weiter:

- alle
- Bug - ein Fehler im System (Exceptions)
- Feature Request - Anfrage, eine Funktionalität zu implementieren
- Funktionalität - eine Funktionalität, die zwar funktioniert, aber anders als gewollt (sie tut etwas, aber das Falsche, wirft aber keine Exception)
- Layout/Look&Feel/CSS - Fehler oder Verbesserungen zur Optik

Die Version gibt an, in welcher Version von Storm der Fehler entdeckt wurde:

- alle
- Alpha Release
- Beta Release

Das Feld *Erstellt von* gibt an, von wem das Ticket eingestellt wurde, hier können alle in der eGroupware vorhandenen Nutzer ausgewählt werden. Die Auswahl von *Erstellt von* zeigt die Tickets an, ohne nach Einstellendem zu filtern.

Weiter wird der Status eines Tickets unterteilt in:

- alle
- nicht geschlossen - alle Status, außer geschlossen
- eigene nicht geschlossen - alle Tickets, die dem eingeloggten Nutzer zugeordnet und nicht geschlossen sind
- geschlossen - Aufgabe des Tickets wurde erledigt
- Review - wurde das Ticket bearbeitet, wird es auf Review gestellt, danach überprüft der Testbeauftragte, ob der Fehler wirklich behoben wurde

Das Feld *Zugewiesen an* gibt an, an wen das Ticket vergeben wurde, hier können einzelne Nutzer oder auch Gruppen ausgewählt werden:

- Storm EE Gruppe - Eingabeeditor
- Storm IF Gruppe - Web 2.0
- Storm SA Gruppe - Systemadministration
- Storm SE Gruppe - Schemaeditor
- Nutzername - ein einzelner Nutzer

Weitere Hinweise zur Ansicht:

- sind in das Suchfeld Stichworte eingetragen, wird danach gefiltert
- die grau-weißen Pfeilschaltflächen blättern in den Tickets (sind leicht zu übersehen)
- um sicherzugehen, dass kein Ticket übersehen wird, Verfolgungsqueue + Kategorie + Version auf *alle*, Status auf *nicht geschlossen*, *Erstellt von* und *Zugewiesen an* auf den obersten Wert stellen und das Suchfeld leer lassen

Um ein Ticket zu bearbeiten, auf die Zusammenfassung klicken.

- Verfolgungs Queue: Die Queue (Deadline) ändern
- Kategorie: die Kategorie ändern
- Version: die Version (in der der Bug aufgetaucht ist) ändern
- Zugewiesen an: zuständige(n) Nutzer/Gruppe(n) ändern
- Priorität: Priorität ändern
- Status: Status ändern (z.B. schließen, wenn behoben)
- Lösung: Art der Lösung und Fortschritt in % ändern
- Beschreibung: eine Beschreibung des Bugs
- Kommentar hinzufügen: den Bug kommentieren (z.B. warum er geschlossen wurde oder Fragen dazu stellen)
- Anhänge Verknüpfungen: Dateianhänge (z.B. Screenshots oder Logs)

10.3. Testberichte

Es folgen die Berichte der Tests zu gegebenen Meilensteinen. Es wurden jeweils zu diesen Meilensteinen wichtige Funktionen getestet.

10.3.1. Testbericht Alpha 13. Mai 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

anonymer Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Flaggen	ja	CSS	8, 60
Suchfeld	nein		
Helpsystem	ja		
Homebutton	ja		
Infocartbutton	ja		
Bericht	ja		
Artikel	ja		
TagCloud	Fehler		61
Vision & Strategie	nein		62
Leitbild & Strategie	nein		62
Richtlinien & Strategie	nein		62
Nachhaltigkeitsbericht & Strategie	nein		62
Über uns & Strategie	nein		62
Sitemap & Strategie	nein		62
Impressum & Strategie	nein		62
AGB & Strategie	nein		62
Kontakt & Strategie	nein		62

Datenschutz & Strategie	nein		62
Social Bookmarks	nein		63
anonyme Ratings	ja		64
anonyme Kommentare	ja		65
Bericht zum Infocart	nein		66
Artikel zum Infocart	ja		67
Infocart löschen	ja		68
registrieren	ja		
Einloggen	ja		69, 96

eingeloggter Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Artikel zu Infocart	nein		70
Meine Daten	Fehler		71
Meine Berichte	Fehler		72
Alle Berichte	ja		
Infocart	ja		
Hilfe	Fehler		73
Logout	ja	nur über oberes Menü	74
Kommentar zu Artikel	Fehler	Fehler nur bei sehr langen Texten	75
Artikel bewerten	ja		
Artikel kommentieren	ja		
Ansicht allgemein		Encoding teilweise falsch	85

Backend

Das Backend für Redakteure erlaubt für die meisten Module CRUD-Operationen, die getestet werden.

Modul	Erstellen	Lesen	Aktualisieren	Löschen	Bemerkung	# im Verfolgungssystem
Artikel			Fehler		nach langen Texten	76, 78
User	ja	ja	ja	ja		
Role	ja	ja	ja	ja		

Rule	ja	ja	ja	ja		
Module	ja	ja	nein	nein		79
Template	ja	ja	ja	ja	Fehler, wenn alle abgewählt sind	80
Sprache	Fehler	Fehler	Fehler	nicht getestet		81, 82 ,83
Report	ja	ja	ja			
Artikel	ja	ja	Fehler	ja		78
Media lokal	ja	ja	ja	ja		
Media extern	nein					
ChangeLog	ja	ja	ja	ja		
SecurityLog	ja	ja	ja	ja		
AccessLog	ja	ja	ja	ja		

Aktionen, die nicht in das CRUD-Schema passen:

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
Report List Aktion für alle	nein		
Report Filter	ja		
Artikel kopieren	Fehler		93
Media alle löschen	nein		
Media Filter	nein		
Media List Sortierung	Fehler	bei Thumbnail	94
User List Sortierung	ja		
Role List Sortierung	ja		
Rule List Sortierung	ja		
Module List Sortierung	ja		
Template List Sortierung	ja		
Sprachen List Sortierung	ja		
Treport List Sortierung	Fehler	bei Modified by	95
Indikatoren List Sortierung	ja		
Bericht List Sortierung	ja		

ChangeLog List Sortierung	ja		
SecurityLog List Sortierung	ja		
AccessLog List Sortierung	ja		

Der Schemaeditor ist eine sehr komplexe Komponente, die einfache Tabelle genügt nicht für die Übersicht, daher eine eigene Tabelle.

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
TReport erstellen	ja		
TReport löschen	nein		84
Treport Liste	ja		
TReport Liste Filter	ja		
Treport Aktion für alle	nein		
TReport Details anzeigen	nein		
TReport Details bearbeiten	ja		
TReport Index anzeigen	ja		92
neue Ebene anlegen	ja		
Ebene bearbeiten	ja		
Ebene löschen	ja		88
Schema importieren	ja		91
Schema importieren undo	ja		
Ebene Indikator hinzufügen	ja/Fehler	vorgefertigte Indikatoren lassen sich hinzufügen, neu erstellte nicht	89
Kategorie anlegen	ja		
Kategorie anzeigen	ja		
Kategorie Name ändern	ja		
Kategorie Indikator hinzufügen	nein		86
Kategorie löschen	ja	nur, wenn keine Indikatoren zugeordnet	
Indikator anlegen	ja		

Indikator anzeigen	ja		87
Indikator anzeigen Filter	ja		
Indikator editieren	ja		
Indikator löschen	ja		

10.3.2. Testbericht 29. Mai 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

anonymer Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Flaggen	ja	CSS	8, 60
Suchfeld	nein		
Helpsystem	nein		
Homebutton	ja		
Infocartbutton	ja		
Bericht	ja		
Artikel	ja	Codierung bei Link Title	85
Social Bookmarks	nein	wegen mangelnder Acconts nur StudiVZ getestet, ausführlichere Tests vor Beta-Release	63
TagCloud	nein		
Vision & Strategie	nein		62
Leitbild & Strategie	nein		62
Richtlinien & Strategie	nein		62
Nachhaltigkeitsberichte & Strategie	nein		62
Über uns & Strategie	nein		62
Sitemap & Strategie	Fehler		62,100
Impressum & Strategie	Fehler		62,101
AGB & Strategie	nein		62,101
Kontakt & Strategie	nein		62,101
Datenschutz & Strategie	nein		62,101
anonyme Ratings	ja		64

anonyme Kommentare	ja		65
Bericht zum Infocart	nein		66
Artikel zum Infocart	Fehler		67,99
Infocart löschen	ja		68
registrieren	ja		
Einloggen	Fehler		69,96,102
Logout		sollte nicht sichtbar sein, da nicht eingeloggt	98

eingeloggter Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Artikel zu Infocart	ja		70
Meine Daten	nein		71
Meine Berichte	nein		72
Infocart	ja		
Hilfe	Fehler		73
Logout	ja	nur über oberes Menü	74
Kommentar zu Artikel	ja	sehr lange Texte lassen sich in Opera nicht anzeigen (kein Scrollen möglich)	75,104
Artikel bewerten	ja		
Artikel kommentieren	ja		

Backend

Das Backend für Redakteure erlaubt für die meisten Module CRUD-Operationen, die getestet werden.

Modul	Erstellen	Lesen	Aktualisieren	Löschen	Bemerkung	# im Verfolgungssystem
Artikel			Fehler		nach langen Texten	76, 78
User	ja	ja	ja	ja	PLZ ist String; User (edit) create Article wirft Exception	103,105,106

Role	ja	ja	ja	ja		
Rule	ja	ja	ja	ja		
Module	ja	ja	nein	nein		79
Template	nein	ja	ja	ja	Fehler, wenn alle abgewählt sind	80,108
Sprache	Fehler	Fehler	Fehler	nicht getestet		81, 82 ,83
Hilfesystem	ja	ja	nein	nein		109,110
Report	ja	ja	ja	nein		
Artikel	ja	Fehler	ja	ja	Lesen Fehler nur bei neuen Artikeln	78,114
Media lokal	ja	nein	ja	ja	ogg funktioniert nicht mehr	115
Media extern	nein					
ChangeLog	ja	ja	ja	ja		
SecurityLog	ja	ja	ja	ja		
AccessLog	ja	ja	ja	ja		

Aktionen, die nicht in das CRUD-Schema passen:

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
Report List Aktion für alle	nein		
Report Filter	ja		
Artikel kopieren	ja		93
Artikel Liste	Fehler	Exception, wenn von Indicator/list/x aus aufgerufen	113
Media alle löschen	nein		
Media Filter	nein		
Media List Sortierung	ja		94
User List Sortierung	ja		
Role List Sortierung	ja		
Rule List Sortierung	ja		
Module List Sortierung	ja		
Template List Sortierung	ja		

Sprachen List Sortierung	ja		
Treport List Sortierung	ja		95
Indikatoren List Sortierung	ja		
Bericht List Sortierung	ja		
ChangeLog List Sortierung	ja		
SecurityLog List Sortierung	ja		
AccessLog List Sortierung	ja		

Der Schemaeditor ist eine sehr komplexe Komponente, die einfache Tabelle genügt nicht für die Übersicht, daher eine eigene Tabelle.

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
TReport erstellen	ja		
TReport löschen	nein		84
Treport Liste	ja		
TReport Liste Filter	ja		
Treport Aktion für alle	nein		
TReport Details anzeigen	ja		
TReport Details bearbeiten	ja		
neue Ebene anlegen	ja		
Ebene bearbeiten	ja		
Ebene löschen	ja		88
Schema importieren	ja		91
Schema importieren undo	ja		
Ebene Indikator hinzufügen	ja		89
Ebene Indikator entfernen	ja		
Kategorie anlegen	Fehler		111
Kategorie anzeigen	ja		

Kategorie Name ändern	ja		
Kategorie Indikator hinzufügen	ja		86
Kategorie löschen	ja	nur, wenn keine Indikatoren zugeordnet	
Indikator anlegen	ja		
Indikator anzeigen	ja		87
Indikator anzeigen Filter	nein		
Indikator editieren	ja		
Indikator löschen	ja		

10.3.3. Testbericht 13. Juni 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

anonymer Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Flaggen	ja	CSS, Flaggen überschneiden sich mit Suchfeld	8, 60
Suchfeld	Fehler	404 Seite	125
Helpsystem	Fehler	404 Seite	125
Homebutton	ja		
Infocartbutton	ja	Encoding falsch auf dem Löschen-Button, hinzufügen von Artikeln nicht getestet wegen Fehler bei Artikel	85
Bericht	ja		
Artikel	nein	Exception durch Rating	85,126
Social Bookmarks	nein	wegen mangelnder Acconts nur StudiVZ getestet, ausführlichere Tests vor Beta-Release	63
TagCloud	nein		
Vision & Strategie	nein		62
Leitbild	nein		62
Richtlinien	nein		62
Nachhaltigkeitsberichte	nein		62
Über uns	nein		62
Sitemap	Fehler	404	62,100
Impressum	ja		62,101
AGB	Fehler		62,101
Kontakt	ja		62,101
Datenschutz	ja		62,101
anonyme Ratings	ja		64
anonyme Kommentare		nicht getestet wegen Fehler bei Artikel	65
Bericht zum Infocart	nein		66

Artikel zum Infocart		nicht getestet wegen Fehler bei Artikel	67,99
Infocart löschen	ja		68
Infocart anlegen	Fehler	Exception	127
registrieren	ja		
Einloggen	Fehler		69,96,102
Logout		sollte nicht sichtbar sein, da nicht eingeloggt	98
Allgemein		Ansicht der Startseite kaputt	124

eingeloggter Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Artikel zu Infocart	Fehler	nicht getestet wegen Fehler bei Artikel	70
Meine Daten	Fehler	Exception bei Edit	71,128
Meine Berichte	nein		72
Infocart		nicht getestet wegen Fehler bei Artikel	
Hilfe	Fehler		73,125
Logout	ja	doppelt	74
Kommentar zu Artikel		nicht getestet wegen Fehler bei Artikel	75,104
Artikel bewerten		nicht getestet wegen Fehler bei Artikel	
Artikel kommentieren		nicht getestet wegen Fehler bei Artikel	

Backend

Das Backend für Redakteure erlaubt für die meisten Module CRUD-Operationen, die getestet werden.

Modul	Erstellen	Lesen	Aktualisieren	Löschen	Bemerkung	# im Verfolgungssystem
Artikel			Fehler		nach langen Texten	76, 78
User	ja	ja	Fehler	ja	Encoding, Exception bei Edit	85,103,105,106,128
Role	ja	ja	ja	ja		
Rule	ja	ja	ja	ja		
Module	ja	ja	nein	nein		79

Template	nein	ja	ja	ja	Fehler, wenn alle abgewählt sind	80,108
Sprache	Fehler	Fehler	Fehler	nicht getestet		81, 82 ,83
Hilfesystem	ja	ja	nein	nein		109,110
Report	ja	ja	ja	nein		
Artikel	ja	Fehler	ja	ja	Lesen Fehler nur bei neuen Artikeln	78,114
Media lokal	ja	nein	ja	ja	ogg funktioniert nicht mehr	115
Media extern	nein					
ChangeLog	ja	ja	ja	ja		
SecurityLog	ja	ja	ja	ja		
AccessLog	ja	ja	ja	ja		

Aktionen, die nicht in das CRUD-Schema passen:

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
Report List Aktion für alle	nein		
Report Filter	ja		
Artikel kopieren	ja		93
Artikel Liste	Fehler	Exception, wenn von Indicator/list/x aus aufgerufen	113
Media alle löschen	nein		
Media Filter	nein		
Media List Sortierung	ja		94
User List Sortierung	ja		
Role List Sortierung	ja		
Rule List Sortierung	ja		
Module List Sortierung	ja		
Template List Sortierung	ja		
Sprachen List Sortierung	ja		
Treport List Sortierung	ja		95

Indikatoren List Sortierung	ja		
Bericht List Sortierung	ja		
ChangeLog List Sortierung	ja		
SecurityLog List Sortierung	ja		
AccessLog List Sortierung	ja		
Einloggen mit falschem Passwort	Fehler		131
Passwort ändern	ja		
(Admin) fremdes Passwort ändern	ja		
SecurityLog:Beim Anmelden, Abmelden und falsches Pass bzw User wird ein SecurityLog erstellt	ja		
AccessLog: Beim öffnen von Artikel wird dieses in AccessLog protokolliert.	nein	nicht getestet wegen Fehler bei Artikel	

Der Schemaeditor ist eine sehr komplexe Komponente, die einfache Tabelle genügt nicht für die Übersicht, daher eine eigene Tabelle.

Aktion	funktioniert	Bemerkung	# im Verfolgungssystem
TReport erstellen	ja		
TReport löschen	nein		84
Treport Liste	ja		
TReport Liste Filter	ja		
Treport Aktion für alle	nein		
TReport Details anzeigen	ja	Encoding	85
TReport Details bearbeiten	ja		

neue Ebene anlegen	ja		
Ebene bearbeiten	ja		
Ebene löschen	ja		88
Schema importieren	ja		91
Schema importieren undo	ja		
Ebene Indikator hinzufügen	ja		89
Ebene Indikator entfernen	ja		
Kategorie anlegen	Fehler		111
Kategorie anzeigen	ja		
Kategorie Name ändern	ja		
Kategorie Indikator hinzufügen	ja		86
Kategorie löschen	ja	nur, wenn keine Indikatoren zugeordnet	
Indikator anlegen	ja		
Indikator anzeigen	ja		87
Indikator anzeigen Filter	nein		
Indikator editieren	ja		
Indikator löschen	ja		
Ebene verschieben	ja		
Schema löschen	ja	nur bei Schemas von denen noch kein Bericht erstellt wurde	
Anzeige Indikatoren einer Ebene	ja		
Export der Indikatoren eines Schemas	ja		

10.3.4. Testbericht 03. Juli 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

anonymer Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Internationalisierung	ja	unvollständig:siehe Kommentare	8
Datenschutz	ja	nur Deutsch	8
Kontakt	ja	nur Deutsch, CSS nicht einheitlich	8,10
AGB	nein	Fehler:404	12
Impressum	ja	nur Deustch, CSS nicht einheitlich	8,10
Sitemap	nein	Fehler:404	12
Newsletter	ja	nur Englisch, Registrierung funktioniert, Layout in Storm einbinden?	8,137
Suche	ja	Sprache umschalten löscht Suchergebnis, findet keine Tags	138
Hilfesystem	nein	keine Hilfe vorhanden, neue Hilfe lässt sich nicht ohne Fehler anlegen, HTML Zeichen in der Anzeige der Hilfe (helparticle/show), Fehlertext01	139
Vision&Strategie	nein	Link auf home Seite	62
Leitbild	nein	Link auf home Seite	62
Richtlinien	nein	Link auf home Seite	62
Nachhaltigkeitsberichte	nein	Link auf home Seite	62
Über Uns	nein	Link auf home Seite	62
Tag Cloud	nein	Link auf home Seite	62
Infocart	Fehler	Fehlertext02 (wenn von Ansicht != Artikel), nur Englisch, Artikel können doppelt hinzugefügt werden, PDF Export verursacht 404 Fehler	8,140

Recommender Engine	ja (Zuletzt betrachtete Artikel)	nur Deutsch, der gleiche Artikel erscheint immer wieder	8
Artikel anzeigen	ja	Sprache uneinheitlich (Deutsch, Englisch gemischt), große Bilder werden nicht korrekt angezeigt (gehen zur Seite raus)	8,10
Indikator anzeigen	ja	CSS der Tagcloud wird zerstört, anonymer Nutzer kann Daten ändern	10,141
Kommentar abgeben	nein	Grails Fehler	142
Social Networks Links		funktioniert nicht: Digg, folkd, Link Arena	143
Ratings	ja		
Registrierung	ja	Sprache umschalten verursacht Fehler 405, Sprache uneinheitlich (Deutsch, Englisch gemischt)	8,144
User Edit	nein	anonymer Nutzer darf nicht editieren können, Fehlertext03	141,145
Einloggen	ja		

eingeloggter Nutzer

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Meine Daten		fehlt	146
Meine Berichte		fehlt	146
Ratings	ja	alle Artikel können bewertet werden, auch neue/nicht freigegebene	147
Infocart		Artikel eines eingeloggten Users sind auch nach dem Ausloggen noch im Infokorb, ruft dann ein anderer anonymer Nutzer den Infokorb auf, gibt es eine Exception, Fehlertext05	148
Recommender Engine	Fehler	wirft ab und zu Exceptions (bisher nicht sicher reproduzierbar), Bilder vor den Einträgen werden nicht angezeigt	149

Bereich Admin

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
User edit	nein	Fehlertext03	141,145
Ausloggen	ja		
Rolle hinzufügen	ja	Sprache uneinheitlich (Deutsch, Englisch gemischt)	8
Rolle bearbeiten	ja	aber Recht entfernen geht nicht	150
Recht hinzufügen	ja		
Recht bearbeiten	ja		
Modul hinzufügen	ja		
Modul bearbeiten	ja	Configs können nicht entfernt werden, Modul kann nicht gelöscht werden	151
Hilfe anlegen	ja		
Hilfe editieren	nein	CSS verhindert die Möglichkeit zum Editieren	10
SAPConnections	nein	Fehler 404	152
BAPI	anlegen, bearbeiten, löschen: ja	Wie Verbindung testen?, Sprache uneinheitlich (Deutsch, Englisch gemischt)	8

Bereich Editor

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Allgemein		nur Deutsch, Überschrift ragt in Textfeld bei Liste	8,10
Bericht erstellen	ja		
Bericht bearbeiten	ja		
Bericht anzeigen	ja	Beschreibung hat HTML-Tags	162
Artikel bearbeiten	ja		
Artikel anzeigen	ja		
PDF Download	Fehler	404	163
Bericht Indikatoren Liste anzeigen	ja		
Bericht Indikatoren Liste einen auswählen	nein	die meisten Indikatoren werden nicht gefunden	164

Bericht anzeigen	nein	nur Bericht wird angezeigt, Artikel sind nicht auswählbar	165

Bereich Logs

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Access Log	ja	lassen sich erstellen, editieren, löschen, Sprache uneinheitlich (Deutsch, Englisch gemischt)	8,167
Security Log	ja	lassen sich erstellen, editieren, löschen, Sprache uneinheitlich (Deutsch, Englisch gemischt)	8,167

Bereich Media

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Media lokal hochladen	ja		
Media bearbeiten	ja		
Media löschen	ja		
Media extern hochladen	nein	fehlende Attribute können nicht eingetragen werden	166

Bereich Schema

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Allgemein		Bezeichnungen nicht einheitlich (Menü), nur Deutsch, CSS (Menü) inkonstistent zum Rest der Anwendung Storm	8,10
TReport anlegen	ja	eingeloggter User ist nicht Standardcreator	153
TReport bearbeiten	ja	kein CSS, doppelt vorhanden (bei List und bei Details bearbeiten)	10,154
Ebene importieren	ja		
Ebene löschen	ja		

Ebene anlegen	ja	eingeloggter User ist nicht Standardcreator	155
Kategorie anlegen	ja		
Kategorie editieren	ja		
Kategorie löschen	ja		
Indikator anlegen	ja	Ersteller wird nicht gespeichert	156
Indikator bearbeiten (von Liste)	Fehler	405	157
Indikator bearbeiten (von Indikator View)	ja		
Aktion für alle Indikatoren	nein		158
Filter	nein		159
Indikatoren zum Schema hinzufügen	Fehler	wird versucht einen neu erstellten Indikator hinzuzufügen, so werden statt dessen zwei alte hinzugefügt, beim herausnehmen analog	160
Ebene verschieben	Fehler	Verschieben einer Ebene führt zu verschiedenen Verhalten: View ohne CSS bei keine Ebene ausgewählt, löschen der Ebene wenn auf sich selbst verschoben, Exception wenn eine Ebene höher verschoben, Fehlertext04	161

10.3.5. Testbericht 24. Juli 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
RSS Feeds	ja	Bei Feeds ohne Bilder werden leere Bilder angezeigt	184

Bereich Admin

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Recommender Engine	ja	zuletzt betrachtete Artikel: der selbe Artikel taucht mehrfach in der Liste auf	192
Datenbankverbindungen	ja	kein richtiger Test, da mir keine Variierungsmöglichkeiten vorliegen	

Bereich Editor

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Fehlende optionale Indikatoren anzeige	ja		
Fehlende (optionale) Indikatoren hinzufügen	Fehler, wenn keine ausgewählt; es werden teilweise falsche Indikatoren hinzugefügt	Fehler01.txt	188, 189, 190

Einfügen von Werten von Indikatoren über den TRE	ja		
live Validierung von Indikatoren	ja		

Bereich Media

Thumbnails	ja	PNG schlechte Farben	183
Einfügen in RTE	nein	wegen Dragn'n'Drop aktuell nicht möglich	

Bereich Web 2.0

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Kommentare abgeben	ja		
Kommentare bewerten	ja		
Kommentare Anzeige	ja	HTML Tags sichtbar, Reihenfolge der Kommentare unklar (scheinbar zufällig)	185, 186
Tags hinzufügen	ja, eingeschränkt	nur 15 Zeichen möglich, Code Injection möglich	187
Tag Cloud	ja		
Suche	Fehler	Fehler beim Suchen	191
Fragesystem	ja		

Bereich Logs
Keine Änderungen.

Bereich Schema

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Indikator Chart	nein	ein Chart wird angezeigt, ist aber unveränderlich	

10.3.6. Testbericht 13. August 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Suchfeld	ja		
Verlinkung Suchergebnisse	ja	führt auf show statt display	211
Infosearch	ja	kein Inhalt	
Sprachumschaltung	ja		
Homebutton	ja		
Infokorb	Fehler	Grails Exception, Fehler01.txt	209
Die beliebtesten Artikel	nein	klappt nicht mehr auf (nur, wenn nicht eingeloggt)	
Last answered Questions	nein	klappt nicht mehr auf (nur, wenn nicht eingeloggt)	
RSS Feeds	nein	klappt nicht mehr auf (nur, wenn nicht eingeloggt)	212
Berichte aller Sprachen anzeigen	ja		
Artikel bewerten	ja	nur ein anonymer Nutzer kann bewerten (der zweite anonyme nicht mehr)	213
Ergebnisse der Tagcloudsuche bewerten	ja	Layout kaputt	215
Tag hinzufügen	ja		
Kommentar schreiben	ja		
auf Kommentar antworten	ja	bei der Antwort wird Titel null angezeigt	214
Kommentar bewerten	ja	nur ein anonymer Nutzer kann bewerten (der zweite anonyme nicht mehr)	
Social Networks		in dieser Version nicht getestet	
Tagcloud	ja	kein CSS mehr für Tags	
Vision&Strategie	ja		

Leitbild	ja		
Richtlinien	ja		
Nachhaltigkeitsberichte	ja		
Über uns	ja		
Newsletter		in dieser Version nicht getestet	
Sitemap	ja		
Impressum	ja		
AGB	ja		
Kontakt	ja		
Datenschutz	ja		
Registrieren	ja	CRUD ist direkt möglich, wirft aber Fehler	216
Login	ja	EMail mit localhost Adresse	217,218
Update last viewed article	ja	kein CSS	222
Zuletzt betrachtete Artikel (Click me)	ja		
Weitere für Sie Interessante Artikel ja			
Die beliebtesten Artikel	ja		
Last answered Questions	ja		
RSS Feeds	ja		

Bereich Admin

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Profil Passowrt ändern	ja		
Profil aktualisieren	Fehler	Fehler02.txt	219
Profil löschen	Fehler	Fehler04.txt	226
letzte Aktionen	ja	Aktionen werden mehrfach angezeigt	220
meine Vorschläge	ja		
User bearbeiten	nein	Fehler02.txt	221
Logout	ja		
RSS Feeds bearbeiten	ja		

Bereich Editor

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Indikator importieren	ja/Fehler	neu angelegte Indikatoren können nicht importiert werden	223
Filter	nein		
Suche	ja		
Artikelvorschau	ja		
SAP/StudIP Verbindung	ja		
Referenzierung von Indikatoren aus dem RTE	ja		
Einbindung Bilder	ja		
Einbindung Flash	nein	verschiedene Fehler	

Bereich Media

Hochladen lokal	ja		
Hochladen extern	ja		
Suche	ja		
Filter	ja/Fehler	Fehler nach Reset, Paginating unsauber	225

Bereich Web 2.0

Alle neuen Funktionen in dieser Version sind im Frontend, siehe 10.3.6.

Bereich Logs

Keine neuen Funktionen in dieser Version.

Bereich Schema

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Indikator anlegen	ja		
Charts für Indikatoren	ja		
Indikator bearbeiten	ja/Fehler	das oberste Bearbeiten wirft einen Fehler, die anderen haben kein CSS	224

10.3.7. Testbericht 28. August 2010

Frontend

Das Frontend für den Standardnutzer besteht hauptsächlich aus Links, die den Nutzer durch die Anwendung führen.
Link/Bereich gibt an, was getestet wurde.
funktioniert gibt an, ob es funktioniert (ja), noch nicht funktioniert, aber auch keinen Fehler wirft, d.h. es folgt in einem späteren Release (nein), oder einen Fehler produziert (Fehler).

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
User registrieren	ja		
User einloggen	ja		
Tags: Groß-/ Kleinschreibung bleibt erhalten	ja		
Leerzeichen in Tags möglich	ja		
Kommentare löschen	ja		
Startseite Suche Autocomplete	nein		239
Startseite allgemein		Menüpunkte an falscher Stelle	238

Bereich Admin

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
User zum Admin machen	nein	Admin-Menü erscheint erst, wenn auch Backend Rolle zugeteilt ist	
User zum Backend-User machen	ja		

Bereich Editor

Link/Bereich	funktioniert	Bemerkung	# im Verfolgungssystem
Report Suche	ja		

Report Filter	ja		
Report Autocomplete	ja	nur, wenn erstes Wort eingegeben	
Artikel Suche	ja		
Artikel Filter	ja		
Artikel Autocomplete	ja	nur, wenn erstes Wort eingegeben	
Anzeige, ob ein Artikel gesperrt ist	ja		
Releaselock (Sperre zurücksetzen)	ja		
Liste der Indikatoren für einzelne Artikel	ja		
Indikatoren Suche	ja		
Indikatoren Autocomplete	ja		
Indikatoren Filter	ja		

Bereich Media

Suche	ja		
Autocomplete	ja		
Filter	ja		
Übersicht für Dateien, wo sie genutzt werden	ja		
neue Formate allgemein	ja	Probleme bei Dateien mit Leerzeichen	243
wmv	ja		
mpg	ja		
avi	ja		
rm	ja		
mov	ja		
ogg	ja		
mp3	ja		
swf			
gif	ja		
jpg	ja		
tif	nein		

Bereich Web 2.0
Alle neuen Funktionen in dieser Version sind im Frontend, siehe 10.3.7.

Bereich Logs
Keine neuen Funktionen in dieser Version.

Bereich Schema
Keine neuen Funktionen in dieser Version.

10.3.8. Testbericht 30. September 2010

TODO: ACHTUNG! Auf Grund der aktuell noch laufenden Testphase kann das erst fertig gestellt werden, wenn die Testphase beendet ist, am 02.10.2010.

Der letzte Testbericht unterscheidet sich von den vorhergehenden. Bisher wurden zu den Meilensteinen hauptsächlich die bis zum jeweiligen Datum neu fertig gestellten Funktionen getestet und dokumentiert, der letzte Test beinhaltet 7 Testsuiten für die Selenium IDE (siehe 10.2.2.2), mit ca. 100 Testfällen und über 1900 Aktionen. Trotz dieser großen Zahl kann der Test nicht als vollständig angesehen werden, das hieße jedes existierende Element zu verifizieren und jeden möglichen Mausklick sowie jede mögliche Eingabe zu tätigen, dies ist allerdings nicht zweckmäßig. Zweckmäßig ist das Testen der zuvor bestimmten Anwendungsfälle im Testszenario (siehe Abschnitte 10.2.2 und 10.2.2.1) um sicherzustellen, dass die Anwendung genau das leistet, was von ihr erwartet wird.
Es folgt eine Übersicht der Testfälle in den Testsuiten und das Ergebnis des jeweiligen Tests. TODO: Ergebnisse am Samstag.

Testfälle

TODO: Sortieren Und zwar NACH dem letzten Testdurchlauf, ausser Olaf fässt das hier keiner an! Fehler

Testsuite Web 2.0 - Testfall und Beschreibung	Erfolg
search - Suche nach Artikeln und Berichten	OK
helpSearch - Suche nach Hilfeartikeln	Fehler
helpArticles - Hilfeartikel ansehen	OK
checkHelpMenue - Menü überprüfen	OK
crudHelpTopic_c - Hilfekategorie erstellen	OK
crudHelpTopic_r - Hilfekategorie lesen	OK
crudHelpTopic_u - Hilfekategoie editieren	OK
crudHelpTopic_d - Hilfekategorie löschen	OK
crudHelpArticle_c - Hilfeartikel erstellen	OK
crudHelpArticle_r - Hilfeartikel lesen	OK
crudHelpArticle_u - Hilfeartikel editieren	OK
crudHelpArticle_d - Hilfeartikel löschen	OK
removeUserAuthor - Nutzer die Redakteurrolle entziehen	OK
addToInfocart - Artikel in den Infokorb legen	OK
showInfocart - Infokorb anlegen	OK
removeFromInfocart - Artikel aus dem Infokorb löschen	OK
newInfocart - neuen Infokorb anlegen	OK
infocartOnePage - Artikel im Infokorb auf einer Seite Anzeigen	OK
infocartToPDF - Artikel im Infokorb als PDF anzeigen	OK
loadInfocart - Infokorb laden	OK
editInfocart - Infokorb editieren	Fehler
delInfocart - Infokorb löschen	Fehler
otherInfocarts - Infokörbe anderer Nutzer anschauen	OK
userInfocart - Profil von Nutzern anschauen, die einen Infokorb freigegeben haben	OK
registerUser - Registrieren	OK

Testsuite Systemadministration - Testfall und Beschreibung	Erfolg
login - Einloggen	OK
logout - Ausloggen	OK
editProfile - Profil editieren	OK
lastViewedArticles - zuletzt angesehene Artikel	OK
latestRecommendations - aktuelle Empfehlungen	OK
crudUser_c - Nutzer erstellen	OK
crudUser_r - Nutzerdaten lesen	OK
crudUser_u - Nutzer editieren	OK
crudUser_d - Nutzer löschen	OK
crudRole_c - Rolle erstellen	OK
crudRole_r - Rolle lesen	OK
crudRole_u - Rolle editieren	OK
crudRole_d - Rolle löschen	Fehler
crudRule_c - Recht erstellen	OK
crudRule_r - Recht lesen	Fehler
crudRule_u - Recht editieren	Fehler
crudRule_d - Recht löschen	Fehler
logs - Log Dateien lesen	OK

Testsuite Schemaeditor - Testfall und Beschreibung	Erfolg
makeUserAuthor - Nutzer zum Redakteur befördern	OK
createcategory - Kategorie erstellen	OK
createIndicator - Indikator erstellen	OK
createGuideline - Richtlinie erstellen	OK
editCategory - Kategorie editieren	OK
editIndicator - Indikator editieren	OK
editGuideline - Richtlinie editieren	OK
createScheme - Schema erstellen	OK
editScheme - Schema editieren	Fehler
exportSchemeList - Schemaliste exportieren	OK
search - Suche nach Schemata	OK
delSchemes - Schema löschen	OK
importLayer - Ebene importieren	OK
moveLayer - Ebene verschieben	OK
showArticleAsPDF - Artikel als PDF anzeigen	OK
showInfocartAsPDF - Infokorb als PDF anzeigen	OK
showReportAsPDF - Bericht als PDF anzeigen	OK
showDiagramm - Indikator mit Diagramm anzeigen	OK
validateReport - Bericht validieren	OK
articleXML - Artikel nach XML exportieren	OK
reportXML - Bericht nach XML exportieren	OK
indicatorXML - Indikator nach XML exportieren	OK
TNodeXML - Schema nach XML exportieren	OK
TReportXML - Berichtstruktur nach XML exportieren	OK
delScheme - Schema löschen	OK

delSchemeGuideline - Richtlinie löschen	OK
delSchemeIndicator - Indikator löschen	OK
delSchemecategory - Kategorie löschen	OK

Testsuite Eingabeeditor - Testfall und Beschreibung	Erfolg
createUploadMedia - Dateien hochladen	OK
createDownloadMedia - Dateien von Webquellen laden	OK
createPaginate - genug Dateien laden, um mehrere Seiten zu erzeugen	OK
searchBase - in Dateien suchen	OK
createReport - Bericht erstellen	Fehler
searchReport - in Berichten suchen	OK
readReport - Bericht lesen	OK
editReport - Bericht editieren	OK
closeReport - Bericht schließen	OK
copyReport - Bericht kopieren	Fehler
searchArticle - Artikel suchen	OK
readArticle - Artikel lesen	OK
publishReport - Bericht freigeben	OK
checkArticles - Aktionen der Artikel testen	OK
delMedia - Dateien löschen	OK

Testsuite Nutzer Rechte
Diese Testsuite testet die in Kapitel **TODO:** ref auf tab:rechte aufgelisteten Rechte für die Nutzergruppen anonyme Nutzer, registrierte Nutzer, Redakteure und Administrator. Alle dort aufgelisteten Rechte wurden erfolgreich getestet und für korrekt befunden.

11. Zusammenfassung und Ausblick

Die Projektgruppe Sustainable Online Reporting Model hat von September 2009 bis einschließlich September 2010 an der Fakultät II der Carl von Ossietzky Universität ein System entwickelt, mit dessen Hilfe eine Nachhaltigkeitsberichterstattung im WEB 2.0 ermöglicht wird. Die Idee eines Systems zur Nachhaltigkeitsberichterstattung mit dem Fokus der direkten Integration der Öffentlichkeit respektive der Stakeholder wurde aufgegriffen und mit neuen Technologien und Features, speziell mit den Errungenschaften des WEB 2.0, aufgewertet und in Bezug auf Dialogorientierung und Usability optimiert.

Nachdem das Web 2.0 - bzw. die Errungenschaften des Web 2.0 - in der Software STORM integriert wurden, wäre eine Weiterentwicklung der Software hinsichtlich des Web 3.0 denkbar. Unter Web 3.0 wird das Semantic Web verstanden, also eine Spezialisierung des Semantischen Netzes. Das Ziel des Semantic Web ist es, die Bedeutung von Informationen für den Computer verwertbar zu machen. Es soll also die Qualität vorhandener Informationen auf semantischer Ebene verbessert werden. (Siehe [Unkg].) Durch die Integration des Web 3.0 Konzeptes könnten die Informationen in den Artikeln computerbasiert ausgewertet und qualitative Verbesserungen angestoßen werden. So könnte beispielsweise nicht nur geprüft werden, ob ein Indikator verwendet wird, sondern auch ob die Ausprägung des Indikators semantisch korrekt ist.
Auch die Implementierung einer Auswertung von Benutzerprofilen und Infokörben stellt eine mögliche Erweiterung dar. So könnten spezifische Analysen gebildet werden um die gezielt Informationen zu den Benutzern und deren Verhalten zu ermitteln. Gerade wenn ein Unternehmen die Software STORM einsetzt, könnte mit der Information, welches Publikum angesprochen wird, die Nachhaltigkeitsberichterstattung noch gezielter ausgerichtet werden. Diese Informationen könnten auch im Bereich Marketing weiter verwendet werden.

Des weiteren ist eine Erweiterung von STORM hinsichtlich der Unterstützung der Extensible Business Reporting Language (XBRL). XBRL ist eine auf XML basierende Sprache, mit der Berichte computergestützt erzeugt und verarbeitet werden können. XBRL wird derzeit vor allem für die Finanzberichterstattung genutzt und ist in diesem Sektor der Berichtserstattung bereits ein anerkannter Standard. In der Thematik der Nachhaltigkeitsberichterstattung wird ein erster Ansatz von der Global Reporting Initiative erprobt, der eine nachhaltigkeitsorientierte Berichterstattung in elektronischer Form erlauben soll. Mit dieser Erweiterung könnte STORM zukunftsweisend gestaltet sein. Kooperationen mit der GRI zur Erprobung des praktischen Einsatzes sind denkbar.

Im Bereich der Systemadministration könnte die Softwaren in der Komponente Benutzerverwaltung eine Erweiterung hinsichtlich der Verwendung von OpenID erfahren. OpenId ist ein dezentrales Authentifizierungssystem für Webseiten. Ein Benutzer registriert sich beim OpenID-Anbieter einmal mit Benutzername und Passwort. Dabei bekommt der Benutzer eine OpenID-Identität. Mit dieser OpenID-Identität kann sich der Benutzer bei jeder Open-ID-unterstützenden Webseite ohne Registrierung anmelden. (Siehe [Fouc].) Auch die STORM-Software könnte eine Open-ID-unterstützende Software werden.

Die Startseite der Anwendung könnte dynamischer gestaltet sein. Die Inhalte der Startseite - u.a. die einzelnen Ergebnisse des Vorschlagswesens, der einführende Text, die TagCloud

- könnten durch den Endbenutzer konfigurierbar sein. Der Benutzer könnte bestimmte Inhalte ein- oder ausblenden, oder die Inhalte nach seinen Wünschen auf der Startseite positionieren. So könnte die Startseite dynamisiert werden.

Der Eingabeeditor könnte soweit erweitert werden, als dass es eine umfassende Importfunktion gibt. Vorgefertigte Textbausteine - die beispielsweise in MS Word oder Open Office geschrieben wurde und im entsprechenden Format vorliegen - könnten mit der Importfunktion direkt in den Eingabeeditor geladen werden.

Die Software STORM könnte zudem um eine Komponente zur direkten Unterstützung der Erstellung von Templates erweitert werden. Die Erstellung von Templates, also Vorlagen zum Design der Anwendung, könnten besser unterstützt und damit auch für weniger erfahrene Benutzer verfügbar gemacht werden.

Denkbar wäre auch die Einbindung eines Umfragesystems in die Software STORM. So könnten die Mitglieder der STORM einsetzenden Organisation Fragestellungen zu dem Prozess der Nachhaltigkeitsberichterstattung oder anderer Art direkt im System geklärt werden.

12. Fazit der Projektgruppe

Ein Projekt mit der Dauer eines Jahres und entsprechendem Umfang durchzuführen, stellte für alle zwölf PG-Mitglieder eine Herausforderung dar. Im Ablauf des Projektes wurden viele Entscheidungen demokratisch getroffen. Vor allem aber die Zusammenarbeit der vielen verschiedenen Persönlichkeiten und verschiedenen Kenntnisstufen stellte eine Herausforderung dar. Die PG-Mitglieder haben gelernt, in einem großen Projektteam zu agieren. Gerade ein optimaler Informationsfluss und eine ausgeprägte Kommunikationsstrategie stellten sich als essenziell heraus. Die Aufteilung in die Entwickergruppen und die Etablierung der Rollen waren die Grundlage für den guten Ablauf der PG.

Einige Entscheidungen stellten sich - nachdem diese einige Zeit im Einsatz waren - als optimierungsbedürftig heraus. Die zu Beginn der PG beschlossene Zeiterfassung wurde nach 3 Monaten wieder abgeschafft. Es stellte sich heraus, dass die Zeiterfassung einen recht hohen Aufwand beim einzelnen PG-Mitglied erzeugt und die Auswertungen nur begrenzt im Rahmen einer Fortschrittskontrolle der Softwareentwicklung einsetzbar sind. Nach der Abschaffung der Zeiterfassung wurde ein Meilensteincontrolling eingeführt. In Bezug auf die Vorgängen, die zur Erreichung eines bestimmten Meilensteins notwendig sind, wurde von den Entwicklern der gefühlte Fertigstellungsgrad wöchentlich gemeldet und eine Abschätzung gegeben, ob dieser Vorgang termingetreu fertig gestellt werden konnte. Im Falle von auftretenden Komplikationen konnte auf Basis dieses Meilensteincontrollings sofort reagiert werden und wichtigere Vorgänge mit mehr Ressourcen termingerecht abgeschlossen werden. Als Fazit sieht die PG hier, dass ein Controlling nicht auf der reinen Anzahl von geleisteten Stunden für bestimmte Vorgänge, sondern auf der Basis von Fertigstellungsgrades erfolgen sollte.

Die PG stellte sich im Laufe des Jahres auch besonderen Herausforderungen. So wurde beispielsweise die Alpha-Präsentation aufgrund einer terminlichen Änderung auf Kundenseite kurzfristig verschoben. Eine weitere Herausforderung war die Verfügbarkeit des vom Kunden bereitgestellten Servers. Aufgrund der hohen Flexibilität und Kundenorientierung innerhalb der PG konnten die PG-Mitglieder mit diesen besonderen Herausforderungen gut umgehen. An dieser Stelle war die Projektgruppe eine sehr gute Vorbereitung auf das wirkliche Arbeitsleben nach dem Studium.

Im Nachhinein war die Projektgruppe eine sehr gute Erfahrung. Neben der Verbesserung der sozialen Kompetenzen bei jedem Einzelnen wurden auch die fachlichen Kompetenzen (z.B. Erlernen der Programmiersprache Groovy) erweitert. Neben einer nicht unwesentlichen Menge an Stress, die mit dieser Projektgruppe einher ging, hat die Projektgruppe auch viel Spaß bereitet. In dem Team der PG konnten neue Freunde gefunden werden und auf Sozialabenden konnten schöne Debatten geführt werden.

Abschließend bleibt festzuhalten, dass die Durchführung einer solchen Projektgruppe sehr sinnvoll ist, da die Studierenden vielfältige Erfahrungen sammeln und gerade durch die „Auftraggeber-Auftragnehmer-Beziehung“ zu den Betreuern sehr gut auf den Einstieg nach dem Studium vorbereitet werden.

Teil V.

Anhang

A. Installationsanleitung

Diese Installationsanleitung beschreibt neben der Installation von Storm auch die Anpassung und Erweiterung der Software.

A.1. Voraussetzungen

Diese Anleitung ist speziell auf das Betriebssystem Ubuntu in der Version 10.04 Desktop ausgelegt. Alle Befehle, Pfadangaben und Softwarepakete sind für dieses Betriebssystem erstellt. Alternativ können Sie die Anleitung unter einem anderem Betriebssystem wie z.B. Windows XP verwenden. Dazu müssen die Befehle abgewandelt und evt. die Softwarepakete für das richtige Betriebssystem aus dem Internet heruntergeladen werden.

Folgende Softwarepakete müssen auf ihrem Computer installiert werden:

- Java 1.6 SDK
- Grails 1.2.2
- Mysql 5
- Jboss 5
- unzip

Installieren Sie Java 6 SDK durch folgenden Befehl und folgen Sie den weiteren Installationsschritten:

apt-get install sun-java6-jdk

Installieren Sie Mysql 5 durch folgenden Befehl und folgen Sie den Installationsschritten: Apt-get install mysql-server Im nächsten Schritt müssen die Datenbanken für Storm angelegt werden. Geben Sie dafür nacheinander folgende Befehle ein, um die Datenbanken für die verschiedenen Environments anzulegen und die dazugehörigen Benutzer zu erstellen.

sudo mysql

Geben Sie nun folgende SQL Befehle in die MySQL Konsole ein:

create database stormDev; grant all on stormDev.* to stormDev@localhost identified by 'daNVqWKVVSxjrxGn';

create database stormTest; grant all on stormTest.* to stormTest@localhost identified by 'jDSrRjryAdqx54cR';

create database stormProd; grant all on stormProd.* to stormProd@localhost identified by 'PEtCcsv2UYdx9qmf';

Grails 1.2.2 befindet sich auf der Installations-DVD. Bitte entpacken Sie diese nach ~/grails/grails-1.2.2 mit dem Befehl:

unzip /media/cdrom/installation/grails-1.2.2.zip -d ~

Nach dem Entpacken sollte sich das „bin“ Verzeichnis von „grails-1.2.2“ in „~/grails-1.2.2/bin“ befinden. Führen Sie im Folgenden die weiteren Installationsschritte, beschrieben in ~/grails/grails-1.2.2/README, aus.

Eine vorkonfigurierte Version des Jboss befindet sich auf der Auslieferungs-DVD. Diese Version ist bereits zum Ausführen von Storm vorkonfiguriert und kann direkt benutzt werden, um Storm auszuführen. Bitte entpacken Sie die Datei jboss.tar.gz nach ~mit den Befehlen:

mv /media/cdrom/installation/jboss.tar.gz ~tar xfvz ~/jboss.tar.gz

Zusätzlichen muss der Jboss in die Variable PATH eingetragen werden Öffnen Sie zuerst die .bashrc mit dem Editor:

editor ~/.bashrc

Fügen Sie am Ende der Datei folgende Zeilen ein:

export JBOSS_HOME~/jboss-5.1.0.GA export PATH$PATH:$JBOSS_HOME/bin

Installieren Sie unzip durch apt-get install unzip.

A.2. Storm erweitern

In diesem Kapitel werden anhand zweier Szenarien die Erweiterungsmöglichkeiten von Storm erklärt. Im ersten Szenario wird die Startseite der Applikation verändert, im Zweiten eine neue Komponente an die Startseite angehängt. Als Vorbedingungen muss zuerst der Quellecode auf ihren Computer installiert und ausführbar gemacht werden. Dieses wird im nächsten Unterkapitel beschrieben.

A.2.1. Entwicklungsumgebung einrichten

Kopieren Sie den Ordner code von der Installations-DVD mit allen Unterordnern nach ~/Storm/. Öffnen Sie eine Shell und navigieren Sie nach ~/Storm/code/Trunk/Storm. Dieses Verzeichnis enthält den gesamten Sourcecode der Applikation Storm im grails-typischen Schema. Der Aufbau einer Grailsapplikation wird in den Seminararbeiten XXX und XXX erklärt. Um Änderungen an den Sourcedateien vorzunehmen, benutzen Sie bitte einen Texteditor.

Um die späteren Änderungen an Storm direkt testen zu können, empfiehlt es sich den von grails integrierten WEB Server zu benutzen. Starten Sie dazu eine neue Shell und navigieren Sie erneut nach ~/Storm/code/Trunk/Storm. Jetzt führen Sie den Befehl grails run-app aus. Dieser Befehl führt die Applikation Storm im Entwicklungsmodus aus und stellt diese unter der Adresse http://localhost:8080/Storm/ zur Verfügung. Dieser Prozess kann ca. 10 Minuten dauern. Starten Sie nun einen Web Browser und navigieren Sie auf die oben stehende Adresse. Bitte lassen Sie die beiden Shell Fenster geöffnet. Mit dem ersten Fenster können Sie Änderungen an Storm vornehmen, im Zweiten sehen Sie die Debug Konsole des Web Servers. Den Server können Sie durch die Tastenkombination STRG + C beenden und im Anschluss, wie im Text beschrieben, mit grails run-app starten. Wichtig dabei ist, dass bei jedem Start alle Datenbankeinstellungen auf den Auslieferungszustand zurückgesetzt werden.

A.2.2. Szenario1: Startseite verändern

In diesem Szenario soll die bestehende Startseite abgeändert werden. Damit Sie die Änderungen direkt testen können, muss der grails- integrierte Web Server aus dem vorherigen Unterkapitel aktiv sein.

Öffnen Sie zuerst den View der Startseite mit folgendem Befehl:

editor grails-app\views\welcome\welcomePage.gsp ()

Suchen Sie die Zeile mit dem Text:
<h1><g:message code="welcome.welcomeMsg" default="Welcome to STORM" /></h1>
Diese Zeile enthält den Begrüßungstext, der eigentliche Text ist in den Internationalisierungsdateien ausgelagert. Fügen Sie eine Zeile unter der gesuchten hinzu und schreiben Sie <h2> Hello World</h2> hinein. Im nächsten Schritt speichern und beenden Sie den Editor. Sollte der Grails Web Server aus dem vorherigen Unterkapitel noch aktiv sein, erkennt dieser automatisch, dass sich der View verändert hat und kompiliert diesen erneuert. Ansonsten starten Sie den Web Server, wie es im vorherigen Unterkapitel beschrieben ist. Rufen Sie nun die Adresse http://localhost:8080/Storm/ mit einem Web Browser auf. Unter dem Begrüßungstext sollte „Hello World" stehen.

A.2.3. Szenario2: Komponente erstellen

In diesem Szenario wird eine Telefonbucherweiterung für die Applikation Storm erstellt und in die Startseite integriert. Zusätzlich soll Storm an- und abschaltbar sein. Für diese Erweiterung kann keine besehende Komponente von Storm verwendet werden, da diese Funktionalität bisher nicht besteht. Dementsprechend muss es komplett neu entwickelt werden. Dieses beinhaltet folgende Teile:

- Datenspeicherung
- Logik
- Ausgabe
- Möglichkeit zum an- und abschalten.

Diese Punkte werden in den folgenden Absätzen behandelt.

Die Telefonbucherweitung soll pro Eintrag den Namen und die Nummer speichern. Dieses wird in Grails durch eine Domain Class bewerkstelligt. Erstellen Sie diese mit folgendem Befehl: Grails create domain-class demo.Telefonbuch Dieser Befehl erstellt die Datei grails-app \domain \demo \Telefonbuch. Öffnen Sie diese mit dem Editor und fügen Sie folgenden Code vor static constraints ein:

String Name Integer Telefonnummer

Speichern Sie die Datei.

Auf Basis der Domain Class kann Grails selbstständig die CRUD Operationen erstellen. CRUD steht dabei für create, read, update und delete. Geben Sie dazu folgendes ein, um das Logikelement, den Controller, zu erstellen:

grails generate-controller demo.Telefonbuch

Die dazugehörigen Ausgaben oder auch Views genannt, werden mit folgendem Befehl erstellt:

Grails generate-view demo.Telefonbuch

Ab diesem Schritt kann die Telefonbucherweiterung benutzt werden. Dazu stoppen und starten sie vorsichtshalber den Grails Server, wie beschrieben in XXX, neu. Rufe Sie nun http://localhost:8080/Storm/Telefonbuch auf. Daraufhin werden Sie auf die Startseite umgeleitet und eine Berechtigungsfehlermeldung wird angezeigt. Durch den modularen Aufbau von Storm wurde automatisch erkannt, dass eine neue Erweiterung erstellt wurde. Neue Erweiterungen besitzen standardmäßig keine Rechtezuweisung und können nur von der Rolle Administrator aufgerufen werden. Melden Sie sich mit dem Benutzer Admin und dem Passwort 123456 an und rufen Sie die Adresse http://localhost:8080/Storm/Telefonbuch erneut auf. Jetzt sollten Sie ein Grundgerüst sehen, mit dem Sie die oben erwähnten CRUD Operation durchführen können. Die Telefonbuchverwaltung sollte nicht nur von Administratoren, sondern von allen Rollen benutzt werden können. Aus diesem Grund müssen Sie die Rechteverwaltung anpassen. Als erstes muss ein neues Recht erstellt und im Anschluss den richtigen Rollen zugewiesen werden. Öffnen Sie die Adresse http://localhost:8080/Storm und melden sich erneut als Admin an, falls Sie abgemeldet wurden. Klicken Sie im admin Menu auf “Module verwalten“. Suchen Sie den Eintrag mit dem Namen “Telefonbuch“ und klicken Sie auf die “ID“. Im folgenden Fenster klicken Sie bitte auf “Bearbeiten“ und anschließend auf “Recht(XXX evt Rule) hinzufügen“. Geben Sie in diesem Fenster folgende Werte ein: Name: Telefonbuch_List Actionname: list Der “Actionname“ muss kleingeschrieben werden und genau dem Namen der Action entsprechen. Im nächsten Schritt klicken Sie auf Anlegen und haben somit ein Recht erstellt, welches den Zugriff auf die Action List für das Module Telefonbuch regelt. Folgend muss dieses Recht der Gruppe Anonym und Benutzer zugeordnet werden, damit alle Mitglieder dieser Rollen dieses Recht besitzen. Klicken Sie im Menupunkt “Admin“ auf “Rollen verwalten“. Als nächstes klicken Sie auf die ID mit dem Namen Anonym und im folgenden Fenster auf “Bearbeiten“. Jetzt befinden Sie sich im Bearbeitungsfenster für die Rolle Anonym. Suchen Sie in der Combobox “Rules“ den Eintrag “Telefonbuch Telefonbuch_List“ und klicken Sie diesen mit gedrückter “STRG Taste“ an. Damit haben Sie der Rolle ein neues Recht zugewiesen. Klicken Sie nun auf Speichern oder im Fehlerfall auf “Role Liste“ um den Vorgang abzubrechen. Wiederholen Sie den Vorgang für die Rolle mit dem Namen User. Melden Sie sich jetzt ab und öffnen Sie erneut die Adresse: http://localhost:8080/Storm/Telefonbuch. Jetzt sollten Sie die Seite öffnen können. In diesem Status ist die Erweiterung noch nicht in Storm integriert. Dazu rufen Sie die Datei grails-app\views\welcome\welcomePage.gsp mit einem Editor auf und fügen, wie in Senario 1, unter der Begrüßung folgenden Code ein:

⟨

g:include controller=“telefonbuch“ action=“list“

⟩

⟨

/g:include

⟩

Dieser Code ruft den Controller Telefonbuch mit der Action List auf. Die Action List stellt alle Telefonbucheinträge in einer Liste dar. Speichern Sie die Datei und rufen Sie die Startseite von Storm, durch folgenden Aufruf, auf: http://localhost:8080/Storm/ Direkt

unter dem Begrüßungstext sollten Sie eine Liste aller Telefonbucheinträge finden, welche wahrscheinlich leer ist. Durch die derzeitige Rechtvergabe können nur Administratoren Einträge ändern, löschen, erstellen oder die Details bearbeiten. Allein die Operation List kann von allen Benutzern aufgerufen werden. Im letzten Schritt soll die Erweiterung deaktiviert werden und nicht mehr aufrufbar sein. Hierzu melden Sie sich wieder als Admin an und rufen 'Module verwalten' aus dem Adminmenü auf. Klicken Sie dort auf den Eintrag mit Namen Telefonbuch und im folgenden Fenster auf "Bearbeiten". Hier setzen Sie das Attribut "detachable" von 1 auf 0 und betätigen Sie mit "Speichern". Mit dieser Einstellung haben Sie das Module deaktiviert. Durch ein erneutes Aufrufen der Startseite sollten Sie die Telefonbucherweiterung nicht mehr sehen. Auch ein direkter Aufruf wird die Erweiterung nicht öffnen.

A.3. WAR Container erstellen

War Dateien oder War Ordner werden benutzt, um Java EE Applikationen auf einem Web Server zu installieren. Im Auslieferungszustand befindet sich bereits ein War Ordner, welcher benutzt werden kann, um Storm auf einen Java EE Server, wie dem Jboss, zu installieren. Sollten Sie aber Änderungen vorgenommen haben, wie im vorherigen Kapitel beschrieben, müssen Sie eine neue War Datei erstellen. Dazu öffnen Sie eine Shell und navigieren Sie zu ~/Storm/code/Trunk/Storm und führen folgenden Befehl aus: grails prod war Storm.war Mit diesem Befehl erstellen Sie ein war File für das Produktivsystem. Entpacken Sie das war file nach ~/Storm.war durch folgenden Befehl: unzip target/Storm.war -d ~/Storm.war Löschen Sie im Anschluss zwei nicht benötigte Libs mithilfe der zwei folgenden Befehlen:

rm ~/Storm.war/WEB-INF/lib/xercesImpl-2.7.1.jar

und

rm ~/Storm.war/WEB-INF/lib/xercesImpl.jar

Nun kann der War Ordner auf dem Jboss deployed werden.

A.4. Storm auf dem Jboss ausführen

Storm wurde auf dem Java EE Server JBoss mit der Version 5.1.0.GA getestet. Im Prinzip sollte Storm auf jedem Java EE Server funktionieren. Im Auslieferungszustand wird diese Version mitgeliefert und im Kapitel XXX wurde dieser ins Verzeichnis ~/Storm/jboss/jboss5 installiert. Bevor der Server gestartet wird, beenden Sie bitte den Grails Server aus dem Kapitel XXX. Beide Applikationsserver starten per default auf dem gleichen Port und würden sich gegenseitig stören. Navigieren Sie in das jboss 5 Verzeichnis in einem Shell Terminal und rufen Sie folgenden Befehl auf:

bin/./run.sh -b0.0.0.0

Dieser Befehl startet den Jboss im Terminal. Nach einer kurzen Wartezeit sollte der Server aktiv und via der Seite http://localhost:8080/ erreichbar sein. Localhost kann dabei durch die IP oder DNS Adresse des Rechners ersetzt werden. Diese Seite ist die Jboss Informationsseite und wird in dieser Anleitung nicht weiter erklärt. Weiterführende Informationen erhalten Sie im Verzeichnis XXXX: Als nächstes muss Storm auf dem Jboss installiert

werden. Dazu verschieben Sie den War Ordner von der Installations-DVD nach ~/Storm/jboss/jboss5/server/default mit dem Befehl: mv /media/cdrom/Storm/XXX ~/Storm/jboss/jboss5/server/default Alternativ können Sie auch den neuerstellten war Ordner aus Kapitel XXX in das obige Verzeichnis verschieben. Jetzt muss der war Ordner von seiner jetzigen Position in den Ordner 'deploy' verschoben werden. Dieses erledigen Sie bitte mit folgendem Befehl:

mv ~/Storm/jboss/jboss5/server/default/Storm.war ~/Storm/jboss/jboss5/server/default/deploy

Theoretisch hätte der war Ordner direkt in den Ordner 'deploy' verschoben werden können. Da der Jboss jedoch das Verzeichnis 'deploy' überwacht, darf der Prozess des Verschiebens nicht zu lange andauern. Nach ungefähr 10 Minuten sollte Storm unter der Adresse http://localhost:8080/Storm erreichbar sein. Um Storm zu deinstallieren muss der Storm Ordner aus dem 'deploy' Verzeichnis gelöscht werden.

A.5. STORM-Templatesystem

Storm bietet neben der Funktionalität rund um Nachhaltigkeitsberichterstattungen auch die Möglichkeit, die Anwendung an das eigene Layout anzupassen. Damit kann das firmen- oder institutseigene Corporate-Design (CD) oder aber auch eine komplett eigene Designvorstellung umgesetzt werden.

A.5.1. Grundsätzliche Aufteilung

Um diese Möglichkeit zu erhalten ist Storm dem Gedanken der Trennung von Layout und Struktur auch im Bereich der designtechnischen Anpassung verschrieben, indem eine Template-Engine entwickelt wurde. Diese kapselt alle Strukturelemente des Designs, denen Funktionalitäten zuzuordnen sind in einzelne Komponenten. Templates werden im Ordner grails-app/views/layouts definiert. Die einzelnen Views der Anwendung inkludieren jeweils nur das main.gsp bzw. das admin.gsp Layout. Anhand des in der Anwendung aktivierten Templates, greift das Layout auf das gewünschte Template zu. Nachstehende Auflistung zeigt die Struktur des Template-Systems im Gesamtkontext der Anwendung:

- STORM
  - grails-app
    - conf
    - controllers
    - domain
    - i18n
    - services
    - taglib
    - utils
    - views
      - view xxx
      - view yyy

            - layouts
            - components
            - _login.gsp
            - _search.gsp
            - _language.gsp
            - ...
            - templates
            - default
            - uniol
            - admin.gsp
            - main.gsp
            - ...
        * ...
    - ...
- web-app
    - css
    - images
    - ...
- ...

ACHTUNG: BEI DER ARBEIT AN EINEM NEUEN DESIGN / CD IN STORM IST ZU BEACHTEN, DASS ALLE GRAILS-DATEIEN, DIE IM ZUSAMMENHANG MIT DEM LAYOUT ENTSTEHEN IN FOLGENDER FORM BENANNT WERDEN MÜSSEN: _DATEINAME.GSP. DURCH DEN UNTERSTRICH WERDEN GRAILS-TEMPLATE-DATEIEN DEFINIERT, DIE PER <G:RENDER> INKLUDIERT WERDEN KÖNNEN.

A.5.2. Bereich 'components'

Alle Strukturelemente, die in der Applikation genutzt werden, repräsentieren die Anzeige bestimmter Storm-Funktionalitäten auf der Webseite von Storm. Diese sind wie bereits erwähnt im Bereich 'components' untergebracht. Soll Grails um bestimmte Funktionalitäten erweitert werden, so müssen sich diese hier wiederfinden, andernfalls kann dieser Bereich unverändert bestehen bleiben. Zusammenfassend enthalten Komponenten zusammenhängende grafische Elemente (z.B. Login-Menü) ohne grafische Formatierungen.

A.5.3. Bereich 'templates'

Storm ist durch Grails an die Sitemesh-Umgebung angepasst. Dadurch lässt sich die grundlegende Struktur einer Seite genau bestimmen. In der Standardanwendung ist die Webseite von Storm in die Bereiche

- Admin

- Footer
- Header
- Main
- Sidebar

aufgeteilt. Jedes Seitenelement findet eine Entsprechung in einer Strukturdatei mit gleichem Namen. Beispielsweise gibt es in der Standardauslieferung die Grails-Datei _footer.gsp. Diese beschreibt die grundlegende Struktur des Footers. In ihr eingebunden sind alle weiteren Strukturelemente der Anwendung, welche aus dem Bereich 'components' stammen. Eine Besonderheit hierbei spielen jedoch die Dateien $_main.gspund_admin.gsp.DiesedienenzurAufnahm$

A.5.4. Bereich 'css'

In diesem Bereich werden alle CSS-Dateien des Designs vorgehalten. Während in den Bereichen 'components' und 'templates' die Struktur aufgebaut wird, wird hier die eigentliche Anzeige mittels verschiedenen Stylesheet-Dateien (alternativ auch in einer einzigen möglich) erstellt.

A.5.5. Bereich 'images'

In diesem Bereich werden alle Bilder des Designs vorgehalten.

A.5.6. Erstellung eines neuen Designs

Um ein neues Design zu erstellen wird empfohlen das Template 'default' (templates>default) zu kopieren, unter einem anderen Namen im Bereich 'templates' zu speichern und dann mittels der CSS-Dateien anzupassen. Dies ist der einfachste Weg um ein neues Design zu erstellen und sicherzugehen, dass jegliche Funktionalität von Storm weiter genutzt werden kann. Die Grundkenntnisse die an dieser Stelle wichtig sind, sind so sehr leicht zu erarbeiten. Eine hilfreiche Internetseite in diesem Zusammenhang stellt die Seite www.css4you.de dar. Soll die vorgeschlagene Struktur nicht beibehalten werden, so sind im ersten Schritt die Dateien _main.gsp und _admin.gsp in einen neuen Ordner im Bereich 'templates' zu kopieren und zu jedem Seitenbereich eine eigene Datei mit dem Namen _DATEINAME.gsp zu erstellen. Es ist dabei jedoch zu beachten, dass bereits erweiterte Kenntnisse im Bereich von HTML und CSS verfügbar sein müssen. Eine Schritt für Schritt-Anleitung zur Erstellung der einzelnen Dateien soll an dieser Stelle nicht gegeben werden. Vielmehr wird darum gebeten, sich an den Default-Dateien zu orientieren. Sobald das Design erstellt ist, muss dieses lediglich noch aktiviert werden. Hierzu ist in den Admin-Bereich der Anwendung Storm zu wechseln. Dort gibt es unter Admin > Templates verwalten die Möglichkeit das neu erstellte Template zu bearbeiten. Hierzu muss es in der Liste der gültigen Templates hinzugefügt werden und kann dann danach aktiviert werden.

B. Benutzerhandbuch

...IRENE...

C. Externe Homepage

Die externe Homepage dient der Präsentation des Projektes STORM nach außen. Über die Homepage können sich Interessierte über das Projekt im Allgemeinen, News und Fortschritte, Rahmenbedingungen sowie das Projekt-Team informieren. Zusätzlich wurden Infos und Screenshots zur Anwendung veröffentlicht und ein Testzugang bereitgestellt.

Die externe Homepage der Projektgruppe STORM ist über die URL http://www.storm.uni-oldenburg.de/ erreichbar. Die Homepage wurde mit der Programmiersprache PHP programmiert. Sie liegt auf dem Server (http://vittel.informatik.uni-oldenburg.de) im Unterverzeichnis /var/www/cms/joomla2.

Jede Seite der Homepage entspricht einer *.php-Datei in dem Root-Ordner der Homepage. Von dort aus werden die entsprechenden Inhalte sowie der Header und Footer aus den Unterordnern /de oder /en inkludiert. Auf diese Weise unterstützt die Homepage die Sprachen deutsch und englisch. Es kann jederzeit zwischen beiden Sprachen gewechselt werden.

Im Unterordner /files befinden sich die Seminararbeiten der Projegtgruppe SEM sowie das Poster und der Flyer der Projektgruppe STORM. Diese Dateien werden dem Benutzer zum Download angeboten Im Unterordner /pic befinden sich Screenshots der Anwendung sowie Bilder der Projektgruppe Im Unterordner /CSS befinden sich die gleichen *.css-Dateien wie im Storm-Projekt, sodass das Design der externem Homepage genau dem Design der Anwendung entspricht Im Unterordner /images befinden sich die für das Design benötigten Bilder

D. Übersicht über die Rechtekonfiguration

Modul	Action	Anonym	Nutzer	Redakteur	Admin
user	logout	nein	ja	ja	ja
user	login	ja	ja	ja	ja
user	createUser	ja	ja	ja	ja
user	forgotPassword	ja	ja	ja	ja
user	changePwdUser	ja	ja	ja	ja
user	showUser	nein	ja	ja	ja
user	editPwdUser	nein	ja	ja	ja
user	editUser	nein	ja	ja	ja
user	updateUser	nein	ja	ja	ja
user	lastArticles	nein	ja	ja	ja
user	myRecommendation	nein	ja	ja	ja
user	validateMail	ja	ja	ja	ja
user	myFollowers	nein	ja	ja	ja
user	authenticate	ja	ja	ja	ja
user	save	ja	ja	ja	ja
user	addFriend	nein	ja	ja	ja
user	index	nein	ja	ja	ja
user	list	nein	ja	ja	ja
user	forgotPassword _SetandSendMail	ja	ja	ja	ja
user	auth	ja	ja	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
infocart	list	nein	ja	ja	ja
infocart	listvisible	ja	ja	ja	ja
infocart	listpublished	ja	ja	ja	ja
infocart	save	ja	ja	ja	ja
infocart	show	ja	ja	ja	ja
infocart	display	ja	ja	ja	ja
infocart	showAsOnePage	ja	ja	ja	ja
infocart	edit	nein	ja	ja	ja
infocart	updatename	nein	ja	ja	ja
infocart	togglepublished	nein	nein	ja	ja
infocart	togglevisible	nein	ja	ja	ja
infocart	delete	nein	ja	ja	ja
infocart	addArticle	ja	ja	ja	ja
infocart	deleteItem	ja	ja	ja	ja
infocart	toPdf	ja	ja	ja	ja

infocart	create	nein	ja	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
help	overview	ja	ja	ja	ja
help	listsections	nein	nein	ja	ja
help	createsection	nein	nein	ja	ja
help	savesection	nein	nein	ja	ja
help	editsection	nein	nein	ja	ja
help	updatesection	nein	nein	ja	ja
help	deletesection	nein	nein	ja	ja
help	createarticle	nein	nein	ja	ja
help	savearticle	nein	nein	ja	ja
help	editarticle	nein	nein	ja	ja
help	save	nein	nein	ja	ja
help	updatearticle	nein	nein	ja	ja
help	deletearticle	nein	nein	ja	ja
help	search	ja	ja	ja	ja
help	listarticles	nein	nein	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
indicator	show	ja	ja	ja	ja
indicator	chart	ja	ja	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
taggable	addTag	ja	ja	ja	ja
taggable	addTags	ja	ja	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
usercomment	create	nein	ja	ja	ja
usercomment	save	nein	ja	ja	ja
usercomment	listarticlecomments	ja	ja	ja	ja
usercomment	display	ja	ja	ja	ja
usercomment	reply	nein	ja	ja	ja
usercomment	delete	nein	nein	ja	ja

Modul	Action	Anonym	Nutzer	Redakteur	Admin
media	thumb	nein	nein	ja	ja
media	list	nein	nein	ja	ja
media	index	nein	nein	ja	ja
media	listThumbs	nein	nein	ja	ja
media	searchFilter	nein	nein	ja	ja
media	searchAJAX	nein	nein	ja	ja
media	create	nein	nein	ja	ja
media	save	nein	nein	ja	ja

media	download	nein	nein	ja	ja
media	show	nein	nein	ja	ja
media	update	nein	nein	ja	ja
media	edit	nein	nein	ja	ja
media	delete	nein	nein	ja	ja
media	actionChecked	nein	nein	ja	ja
media	updateArticles	nein	nein	ja	ja
media	addToView	nein	nein	ja	ja
media	removeFromView	nein	nein	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
article	index	nein	nein	ja	ja
article	copy	nein	nein	ja	ja
article	list	nein	nein	ja	ja
article	searchFilter	nein	nein	ja	ja
article	create	nein	nein	ja	ja
article	save	nein	nein	ja	ja
article	show	nein	nein	ja	ja
article	starsrated	ja	ja	ja	ja
article	display	ja	ja	ja	ja
article	edit	nein	nein	ja	ja
article	update	nein	nein	ja	ja
article	cancel	nein	nein	ja	ja
article	approve	nein	nein	ja	ja
article	preview	nein	nein	ja	ja
article	open	nein	nein	ja	ja
article	close	nein	nein	ja	ja
article	revision	nein	nein	ja	ja
article	addTag	nein	ja	ja	ja
article	addTags	nein	ja	ja	ja
article	findByTag	ja	ja	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
report	index	nein	nein	ja	ja
report	list	nein	nein	ja	ja
report	searchFilter	nein	nein	ja	ja
report	create	nein	nein	ja	ja
report	toPdf	nein	nein	ja	ja
report	xml	nein	nein	ja	ja
report	save	nein	nein	ja	ja
report	cancel	nein	nein	ja	ja
report	validateReport	nein	nein	ja	ja
report	approve	nein	nein	ja	ja
report	open	nein	nein	ja	ja
report	show	nein	nein	ja	ja
report	display	ja	ja	ja	ja

report	edit	nein	nein	ja	ja
report	copy	nein	nein	ja	ja
report	update	nein	nein	ja	ja
report	close	nein	nein	ja	ja
report	autocomplete	nein	nein	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
articleIndicator	index	nein	nein	ja	ja
articleIndicator	list	nein	nein	ja	ja
articleIndicator	allocate	nein	nein	ja	ja
articleIndicator	updateAllocation	nein	nein	ja	ja
articleIndicator	searchFilter	nein	nein	ja	ja
articleIndicator	autocomplete	nein	nein	ja	ja
articleIndicator	searchArticleIndicator	nein	nein	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
category	show	ja	ja	ja	ja
category	list	nein	nein	ja	ja
category	create	nein	nein	ja	ja
category	edit	nein	nein	ja	ja
category	update	nein	nein	ja	ja
category	delete	nein	nein	ja	ja
category	move	nein	nein	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
indicator	show	ja	ja	ja	ja
indicator	list	nein	nein	ja	ja
indicator	create	nein	nein	ja	ja
indicator	edit	nein	nein	ja	ja
indicator	update	nein	nein	ja	ja
indicator	delete	nein	nein	ja	ja
indicator	chart	nein	nein	ja	ja

Modul	**Action**	**Anonym**	**Nutzer**	**Redakteur**	**Admin**
Treport	list	nein	nein	ja	ja
Treport	create	nein	nein	ja	ja
Treport	edit	nein	nein	ja	ja
Treport	update	nein	nein	ja	ja
Treport	delete	nein	nein	ja	ja
Treport	listNodes	nein	nein	ja	ja
Treport	importSchema	nein	nein	ja	ja
Treport	move	nein	nein	ja	ja
Treport	editSchema	nein	nein	ja	ja

E. Seminararbeiten

F. Protokolle

Literaturverzeichnis

[AH03] David Thomas Andrew Hunt. Der pragmatische programmierer. In *Der Pragmatische Programmierer*, page 24. Hanser Fachbuchverlag, 2003. 60

[Ang05] M. Angermeier. The huge cloud lens bubble map web 2.0. http://kosmar.de/archives/2005/11/11/the-huge-cloud-lens-bubble-map-web20/, November 2005. 6, 53

[Bec] Jörg Becker. Grundsätze ordnungsmäßiger modellierung. http://enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Grundsatze-ordnungsgemaser-Modellierung. 57

[Bol06] Dietrich Boles. *Leitfaden zur Durchführung von Projektgruppen*. Dietrich Boles, 2006. 10

[CAE91] G. L. Rein C. A. Ellis, S. J. Gibbs. *Groupware - Some Issues and Experiences*. Communications of the ACM, 34(1), 1991. 21

[Dir09] DirektZu. Direkt zu. http://www.direktzu.de/, 2010 09. 49, 115

[fe] Institut für ökologische Wirtschaftsforschung future e.V. IÖw/future-ranking der nachhaltigkeitsberichte. http://www.ranking-nachhaltigkeitsberichte.de/index.php?id=92. 11

[For07] Peter Forbrig. *Objektorientierte Softwareentwicklung mit UML*. Hanser Fachbuchverlag, 2007. 17, 18

[Foua] Apache Software Foundation. Apache. http://httpd.apache.org/. 22

[Foub] Apache Software Foundation. Subversion. http://subversion.apache.org/. 22

[Fouc] OpenID Foundation. Openid. http://openid.net/get-an-openid/what-is-openid/. 159

[Fow06] Martin Fowler. Continuous integration. http://www.martinfowler.com/articles/continuousIntegration.html, 2006. 19

[GL03] Jeremy York Greg Linden, Brent Smith. Amazon.com recommendations item-to-item collaborative filtering, Januar / Februar 2003. 6, 109, 110, 111

[Góm08] Jorge Marx Gómez. *Internetbasierte Nachhaltigkeitsberichterstattung: Maßgeschneiderte Stakeholder-Kommunikation mit IT*. Erich Schmidt Verlag, 2008. 11

[Gmb] Sun Microsystems GmbH. Mysql. http://www.mysql.de/. 23

[Hat] Red Hat. Hibernate. http://www.hibernate.org/. 23

[HH07] Rolf Isernhagen Hartmut Helmke, Frank Höppner. *Einführung in die Softwareentwicklung*. Hanser Fachbuchverlag, 2007. 17

[MB09] Jochen Ritscher Michael Bächle. Ruby on rails. http://pi.informatik.uni-siegen.de/stt/26_4/03_Technische_Beitraege/03_rubyonrails.pdf, Januar 2009. 60

[Mef92] Heribert Meffert. *Marktorientiertes Umweltmanagement, Grundlage und Fallstudien*. C.E. Poeschel Verlag, 1992. 11

[Mica] Microsoft. Microsoft visio. http://office.microsoft.com/de-de/visio/. 24

[Micb] Microsoft. Msdnaa. http://www.microsoft.com/switzerland/education/de/initiatives/msdnaa.mspx. 24

[Myr] Thomas Myrach. Anforderungsdefinition. http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Anforderungsdefinition. 29

[Nie93] J. Nielsen. *Usability Engineering*. 1993. 6, 18

[Ora] Oracle. Hudson. http://hudson-ci.org/. 24

[Pfr05] Reinhard Pfriem. *Heranführung an die Betriebswirtschaftslehre*. Metropolis Verlag, 2005. 11

[RH] Inc. Red Hat. Jboss. http://www.jboss.com/about/. 23

[Rog08] Holger Rogall. *Ökologische Ökonomie*. VS Verlag für Sozialwissenschaften, 2008. 11

[Spr] SpringSource. Grails ide integration. http://www.grails.org/IDE+Integration. 24

[SR06] James Robertson Suzanne Robertson. *Mastering the Requirements Process*. Addison Wesley, 2006. 29

[tea] The TortoiseSVN team. Tortoisesvn. http://tortoisesvn.net/. 25

[Tho10] Matthias Thoma. Open source lizenzen kurz vorgestellt, 2010. 12

[Tin10] TinyMCE. Tinymce. http://tinymce.moxiecode.com/, 08 2010. 91

[Unka] Unknown. Benutzerverwaltung. http://de.wikipedia.org/wiki/Benutzerverwaltung. 65

[Unkb] Unknown. Browser. http://de.wikipedia.org/wiki/Browser. 60

[Unkc] Unknown. Forum. http://de.wikipedia.org/wiki/Internetforum. 22

[Unkd] Unknown. Grails. http://de.wikipedia.org/wiki/Grails. 23

[Unke] Unknown. Integrierte entwicklungsumgebung. http://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung. 24

[Unkf] Unknown. Object relational mapping. http://de.wikipedia.org/wiki/Object-Relational_Mapping. 23

[Unkg] Unknown. Web 3.0. http://de.wikipedia.org/wiki/Semantisches_Web. 159

[Unkh] Unknown. Webserver. http://de.wikipedia.org/wiki/Webserver. 22

[Ver09] Huber Verlag. Über die trackbackfunktion. http://www.lifepr.de/trackback-info/boxid-132596.html, 2009. 6, 53